



日 本 国 特 許 庁  
JAPAN PATENT OFFICE

JP9 20010049  
#4  
2-6-03  
JM  
10/091840  
03/06/02

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日  
Date of Application:

2001年 6月28日

出 願 番 号  
Application Number:

特願2001-196223

出 願 人  
Applicant(s):

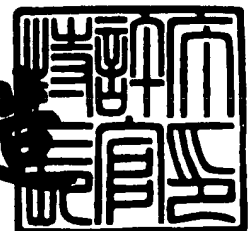
インターナショナル・ビジネス・マシーンズ・コーポレーション

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 7月27日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



【書類名】 特許願

【整理番号】 JP9010049X

【提出日】 平成13年 6月28日

【あて先】 特許庁長官 殿

【国際特許分類】 H04L 1/00

【発明者】

【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 片山 泰尚

【発明者】

【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 森岡 澄夫

【発明者】

【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 山根 敏志

【特許出願人】

【識別番号】 390009531

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【代理人】

【識別番号】 100106699

【弁理士】

【氏名又は名称】 渡部 弘道

【復代理人】

【識別番号】 100110607

【弁理士】

【氏名又は名称】 間山 進也

【先の出願に基づく優先権主張】

【出願番号】 特願2001- 66573

【出願日】 平成13年 3月 9日

【手数料の表示】

【予納台帳番号】 062651

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【包括委任状番号】 0004480

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 復号回路、該復号回路を用いる復号装置、復号方法および半導体デバイス

【特許請求の範囲】

【請求項 1】 符号化された入力デジタル信号が符号のインターリーブ数に対応して並列して入力される入力部と、

前記入力部からのインターリーブされた符号語間でシリアル化されたデータ出力をそれぞれ処理する誤り位置多項式係数算出部と、誤り値多項式係数算出部とを含む処理部と、

前記処理部からの出力と前記入力デジタル信号とが入力され、前記処理部からの出力と、前記入力デジタル信号とから、誤りが訂正された出力デジタル信号をインターリーブ数に対応して並列に出力する出力部とを備える、復号回路

【請求項 2】 前記入力部は、前記入力デジタル信号のシンδροームを算出してインターリーブされた符号語間でシリアル化されたデータとしてシンδροームを前記処理部へと渡し、

前記処理部は、入力される前記シンδροームを使用して、誤り位置多項式係数と誤り値多項式係数とを算出し、

前記出力部は、前記処理部の出力を入力として誤り位置および誤り値を、前記入力デジタル信号からガロア拡大体上における線形演算から生成して前記出力デジタル信号とする、請求項 1 に記載の復号回路。

【請求項 3】 前記入力デジタル信号は、それぞれがリード・ソロモン符号化された  $i$  - チャンネルの並列入力とされ、さらに前記復号回路は、 $i : 1$  または  $1 : i$  のマルチプレクサまたはデマルチプレクサを含む、請求項 1 または 2 に記載の復号回路。

【請求項 4】 前記入力デジタル信号は、光通信における波長多重通信により送信される、請求項 1 ～ 3 のいずれか 1 項に記載の復号回路。

【請求項 5】 少なくとも前記入力部は、順序回路から構成され、前記処理部は、組み合わせ回路から構成される、請求項 1 ～ 4 のいずれか 1 項に記載の復号

回路。

【請求項 6】 デジタル信号の誤り訂正、または暗号化のために使用される、請求項 1 ～ 5 のいずれか 1 項に記載の復号回路。

【請求項 7】 符号化された入力デジタル信号を受信するための入力手段と、前記入力デジタル信号を処理して誤り位置多項式係数と誤り値多項式係数とを算出する処理手段と、前記誤り位置多項式係数と、前記誤り値多項式係数とから誤りが訂正された出力デジタル信号を出力する出力手段とを含み、

前記入力手段は、前記符号化された入力デジタル信号が符号のインターリーブ数に対応して並列に入力され、かつ前記符号化された入力デジタル信号からインターリーブされた符号語間でシリアル化されたデータとしてシンδροームを算出し、

前記処理手段は、前記受信手段から出力されるそれぞれの前記シンδροームを使用して誤り位置多項式係数と誤り値多項式係数とを算出し、

前記出力手段は、それぞれの前記誤り位置多項式係数と前記誤り値多項式係数と、前記符号化された入力デジタル信号とからガロア拡大体上における線形演算により誤りが訂正された出力デジタル信号を符号のインターリーブ数に対応して並列に出力する、復号装置。

【請求項 8】 前記入力デジタル信号は、それぞれリード・ソロモン符号化された  $i$  - チャンネルの並列入力とされ、さらに前記復号装置は、 $i : 1$  または  $1 : i$  のマルチプレクサまたはデマルチプレクサを含む、請求項 7 に記載の誤り復号装置。

【請求項 9】 前記入力デジタル信号は、光通信における波長多重通信により送信される、請求項 7 または 8 のいずれか 1 項に記載の復号装置。

【請求項 10】 少なくとも前記入力部は、順序回路により構成され、前記処理部は、組み合わせ回路から構成される、請求項 7 ～ 9 のいずれか 1 項に記載の復号装置。

【請求項 11】 デジタル信号の誤り訂正、または暗号化のために用いられる請求項 7 ～ 10 のいずれか 1 項に記載の復号装置。

【請求項 12】 デジタル信号を復号するための方法であって、該方法は、

符号化された入力デジタル信号を符号のインターリーブ数に対応して並列して入力するステップと、

前記入力部からのインターリーブされた符号語間でシリアル化されたデータ出力を誤り位置多項式係数算出部と、誤り値多項式係数算出部とを含む処理部を使用してそれぞれ処理するステップと、

前記処理部からの出力と前記入力デジタル信号とを入力し、前記処理部からの出力と、前記入力デジタル信号とから誤りが訂正された出力デジタル信号を生成するステップと、

前記生成された誤りが訂正された出力デジタル信号を、インターリーブ数に対応して並列に出力するステップと

を含む復号方法。

【請求項 1 3】 前記入力ステップは、前記入力デジタル信号のシンδροームを算出してインターリーブされた符号語間でシリアル化されたデータとしてシンδροームを前記処理部へと渡すステップを含み、

前記処理ステップは、入力される前記シンδροームを使用して、誤り位置多項式係数と誤り値多項式係数とを算出するステップを含み、

前記出力ステップは、前記処理部の出力を入力として誤り位置および誤り値を、前記入力デジタル信号からガロア拡大体上における線形演算から生成して前記出力デジタル信号とするステップを含む、請求項 1 2 に記載の復号方法。

【請求項 1 4】 前記入力ステップは、それぞれがリード・ソロモン符号化された  $i$  - チャンネルの並列入力として前記入力デジタル信号を入力するステップを含み、さらに  $i : 1$  のマルチプレクシング・ステップおよび  $1 : i$  のデマルチプレクシング・ステップを含む、請求項 1 2 または 1 3 に記載の復号方法。

【請求項 1 5】 光通信における波長多重通信により送信される入力デジタル信号として入力される、請求項 1 2 ～ 1 4 のいずれか 1 項に記載の復号方法。

【請求項 1 6】 少なくとも前記入力ステップは、順序回路を使用した演算処理を行い、前記処理ステップは、組み合わせ回路を使用して演算処理を行う、請求項 1 2 ～ 1 5 のいずれか 1 項に記載の復号方法。

【請求項 1 7】 デジタル信号の誤り訂正、または暗号化のために使用され

る、請求項 1 2 ～ 1 6 のいずれか 1 項に記載の復号方法。

【請求項 1 8】 デジタル信号を処理するために使用される半導体デバイスであって、該半導体デバイスは、

符号化された入力デジタル信号を受信するための入力手段と、

前記符号化された入力デジタル信号を処理して誤り位置多項式係数と誤り値多項式係数とを算出する処理手段と、

前記誤り位置多項式係数と、前記誤り値多項式係数とから誤りが訂正された出力デジタル信号を出力する出力手段とを含み、

前記入力手段は、インターリーブされた前記符号化された入力デジタル信号がインターリーブ数に対応して並列に入力され、かつ前記符号化された入力デジタル信号からインターリーブされた符号語間でシリアル化されたデータとしてシンδροームを算出し、

前記処理手段は、前記入力手段から出力されるそれぞれの前記シンδροームを使用して誤り位置多項式係数と誤り値多項式係数とを算出し、

前記出力手段は、それぞれの前記誤り位置と前記誤り値と、前記入力デジタル信号とからガロア拡大体上における演算により生成された出力デジタル信号をインターリーブ数に対応して並列に出力する、半導体デバイス。

【請求項 1 9】 前記入力デジタル信号は、それぞれリード・ソロモン符号化された  $i$  - チャンネルの並列入力とされ、さらに前記半導体デバイスは、 $i : 1$  または  $1 : i$  のマルチプレクサまたはデマルチプレクサを含む、請求項 1 8 に記載の半導体デバイス。

【請求項 2 0】 前記符号化された入力デジタル信号は、光通信における波長多重通信により送信される、請求項 1 8 または 1 9 に記載の半導体デバイス。

【請求項 2 1】 前記入力手段は、順序回路から構成され、前記処理手段は、組み合わせ回路から構成される、請求項 1 8 ～ 2 0 のいずれか 1 項に記載の半導体デバイス。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、復号回路、復号装置、復号方法および半導体デバイスに関し、より詳細には、本発明は、高速の光通信分野において特に効果的に誤りを訂正することを可能とする、復号回路、復号装置、復号方法、および半導体デバイスに関する。

## 【 0 0 0 2 】

## 【従来の技術】

## 高速で高度な誤り訂正技術の重要性

インターネットの拡大とe-ビジネスの進展に伴い、コンピュータの扱うデータ量とスピードとは、加速度的に増加している。これに伴い、複数のコンピュータの間におけるデータ通信の速度を高めることが要求されており、近年では40 Gbpsのデータ転送速度を使用する光通信の使用が普及しつつある。このような高速の、特に光通信といった通信方法において、システム・レベルの誤りの発生率を現在と同等に保つためには、コンピュータの扱うデータ量に比例してデータ通信における信頼性をよりいっそう向上させる必要がある。

## 【 0 0 0 3 】

この信頼性を向上させる重要な技術としては、高度な数学を駆使してさまざまな要因（伝送経路でのノイズなど）による誤りを自動的に訂正する誤り訂正符号と呼ばれるものがある。その中でも、今日、多く使われている代表的な誤り訂正符号として、ハミング符号とリード・ソロモン符号を挙げることができる。上述したハミング符号は、基本的にはビット単位の誤り訂正を行なうものであり、その訂正能力は低いといえる。例えば、ハミング符号を使用すれば、1ビットの誤りを発見した場合は訂正するが、2ビットの誤りは検出するのみとなる。ハミング符号を用いる誤り訂正システムは、誤り訂正処理自体が簡単なため、誤り訂正処理を並列的に処理させることにより、1 Gbps（1秒間に10億ビット）を大きく超える高速処理が可能であることが知られている。

## 【 0 0 0 4 】

一方で、リード・ソロモン符号は、複数個の連続したビット単位（シンボル）の誤りを訂正することが可能で高度な訂正能力を有する優れた誤り訂正方法である。しかしながら、リード・ソロモン符号は、処理に複雑な演算を多用するため



並列処理が難しく、例えば、8ビットのデータを100MHzでパイプライン処理を実行させたとしても、800Mb/sの処理速度しか得られないのが現状である。したがって、従来のリード・ソロモン符号の復号方式は、低速の通信分野、またはハードディスクやCD-ROM等の二次記憶装置の分野の製品など、現状では主にデータ処理速度の比較的低い分野で応用されており、今後高速性が要求される分野への応用には限界があった。

## 【 0 0 0 5 】

## 高速光通信分野で要求される誤り訂正技術

特にコンピュータおよびコンピュータが関連するデータ通信に使用される高速光通信の分野では、近年、ますます増大しつつあるインターネットをバックボーンとして、波長多重通信WDM(Wavelength Division Multiplexing)や、さらに波長多重度を向上させたDWDM(Dense WDM)を使った、テラビット／秒の高速通信システムが、所定の長さのフレームを連続して同期転送するSONETといった技術をベースに導入されつつある。

## 【 0 0 0 6 】

光通信を使用する上述したデータ通信における波長多重度の向上に伴い、近接する波長間でのクロストークが問題となる。このようなクロストークに対処するためこれまで、光を用いる波長多重通信における長距離(Long Haul)での転送に際しては、誤り訂正法としてFEC(Forward Error Correction)が使用されている。ITU(International Telecommunication Union)では、ITU-T G.975において、インターリーブされた $m=8$  (8ビット／シンボル)の(255, 239)RS符号 (符号長 $n=255$  バイト)の使用が標準化され、G.709では、FECのFrame構造を定めるDigital Wrapperの標準が採用されている。

## 【 0 0 0 7 】

このような例えばDigital Wrapper標準においては、これまでの低速なシリアル・リード・ソロモン復号回路を複数並列に並べることにより、必要な処理能力が確保されており、リード・ソロモン符号のインターリーブは必要不可欠な技術になっている。

## 【 0 0 0 8 】

## 高速かつ高度な誤り訂正技術における先行技術

これまで、上述したような光通信での必要性とは独立して、組み合わせ回路によるリード・ソロモン符号の平行高速復号の研究が行われてきている。

【0009】

図1は、誤り訂正装置に使用することができるこのような高速復号回路の例を示す。

図1に示される復号回路は、これまでの1復号回路あたりの復号処理速度を10倍以上高速化すること、高度な誤り訂正能力を持つリード・ソロモン符号の誤り訂正処理を、ハミング符号と同程度の処理速度で平行復号を行なうことを実現するものである。図1に示した復号回路では、ガロア体上での対称関数による新たな表現形式を、リード・ソロモン符号の復号処理に応用することにより、誤り値を直接計算できる $O(t)$ 次の誤り値多項式 $E_r(x)$ が用いられている( $t$ は誤り訂正できる最大のシンボル数)。

【0010】

図1に示した復号回路では、この多項式の使用により、シンδροーム計算、誤り位置評価だけではなく、誤り値の評価も、これまでのようにForneyアルゴリズムを用いて2つの多項式の評価結果を除算して求めるのではなく、1つの多項式で直接計算可能とするので、はるかに簡単化することができる。さらに、図1に示した復号回路では、 $E_r(x)$ の係数計算だけではなく、誤り位置多項式 $\Lambda(x)$ の係数計算に対しても、組み合わせ回路に適した表現が採用され、必要な演算回路の数の削減を、高速化と同時に可能としている。

【0011】

図1に示される復号回路を使用することにより、 $0.35\mu m$ の標準的なASIC技術を使って半導体上に試作されたランダムな4バイト誤り訂正処理回路では、320ビット幅のデータを低レイテンシ(45 ns)で並列処理することが可能とされ、現在のシリアル復号回路の典型的な処理速度の800Mb/sの10倍近い7Gb/s(1秒間に70億ビット)以上の処理速度が達成されている。さらに、図1に示した復号回路に対して、大規模並列の誤り訂正処理回路に特化した新たな回路最適化アルゴリズムを使用し、回路共有化手法を用いることにより、回路規模が縮小できること

が示されている。さらに、図1の復号回路は、外部制御回路やレジスタを使わない組み合わせ回路であるので、高速処理にもかかわらず消費電力も抑えることができるという利点がある。

#### 【0012】

しかしながら、図1に示した復号回路においても、光通信に必要とされる40 Gbpsの処理速度には及ばず、また通常の回路共有化手法を用いるだけでは回路規模が、ITUで標準化された8バイト誤り訂正に対応するためには1チップに実装できないほどに大規模になってしまうという問題があった。

#### 【0013】

図2は、従来用いられている低速の復号方式を使用する光通信用誤り訂正回路の概略構成を示した図である。光通信分野での通信速度の向上に伴い、従来の低速のシリアル・リード・ソロモン復号回路を並列に並べる方法は、ますます困難になってきている。既存のRS復号回路では、せいぜい1 Gbpsの処理性能しかない。このため、図2に示した復号方式においては、低速のシリアル・リード・ソロモン復号回路を並列に並べることで必要な処理性能が実現されている。しかしながら、図2に示した従来方式は多くのリード・ソロモン復号回路を並べる必要が生じ、光通信のデータ転送速度に比例して、回路規模が大きくなってしまいうという不都合がある。図3には、図2に示した復号方法を使用する場合の、回路規模と、データ転送速度とをプロットした図を示す。

#### 【0014】

図4には、さらに別の復号回路の従来例を示す(A. Patel, IBM J. Res. Develop., vol. 30 pp. 259-269, 1986)。従来の復号方式では、シンドロームの計算・誤り位置の計算については多項式の評価ですむので、容易に高速化することができる。しかしながら、図4に示すように誤り値の計算でForneyアルゴリズムを用いるため、2つの多項式、すなわち誤り位置多項式の微分 $d\Lambda(x)/dx$ と、シンドロームと誤り位置多項式とから計算される誤り評価多項式 $\Omega(x)$ との評価の後に除算を実行することが必要となる。そして、これが出力の高速化を妨げるクリティカル・パスとなり、十分な高速化を行うことができないという不都合がある。

## 【 0 0 1 5 】

OC-768 SONETでは、復号回路の入出力インターフェースとして、ITU-G 709で決められた16インターリーブを仮定すると、300MHz以上の高速動作が期待されているので、これは、重大な問題となる。図4に示される復号回路を使用し、クリティカル・パスに相当する除算をさらに細かくパイプライン化して出力を高速化しようとすることも試みられている。

## 【 0 0 1 6 】

しかしながら、いくら細かくパイプライン化したとしても図4に示した復号回路では、誤りのない位置でも除算を行なう必要があり、パイプラインの細分化に伴って回路規模・消費電力共に増大する。また、誤り位置にのみ除算を行なおうとすると、誤り位置を事前に計算する必要があり、誤り位置と誤り値の同時計算が行なえないという不都合もある。また、図4に示した復号回路においては、誤りがあるか無いかによって、誤り値の出力に必要なサイクル数が違うので、SONETのような同期式フレームに載せて連続したデータを高速に入出力する必要のある場合には、誤りパターン（誤り数・位置）に依存せず一定のサイクルで高速に誤り値を出力することが困難である。

## 【 0 0 1 7 】

図5にはさらに別の従来の復号回路を示す。図1に示したパラレル・リード・ソロモン復号方式を光通信の分野に応用しようとする、回路あたりの処理性能という観点では既存の方式よりも優れているためインターリーブされていないRS符号では問題なく応用可能である。しかしながら、ITU-T G.975で決められているインターリーブされたリード・ソロモン符号に対しては、高速で大きなバッファとセレクタを用いて信号の順序を並べ替える必要が生じるため、必ずしも効率のよいものではなかった。すなわち、(255, 239)RS符号の符号長は、2040ビットであり、16インターリーブした場合には、16バイト入力255バイト出力のシリアル・パラレル変換回路と255バイト入力16バイト出力のパラレル・シリアル変換回路を必要とし、高速化という点ではある程度の目的を達成することができるものの回路サイズが著しく大規模なものとなり、実用レベルで提供することが困難である。

## 【 0 0 1 8 】

また、上述した復号回路に使用される誤り位置および誤り値の算出においては、ガロア拡大体  $(GF) 2^m$  上における演算を大量に、かつ高速で実行でき、さらには、実装可能な回路規模で処理を実行させることが要求される。従来、上述したガロア体における演算について従来行われてきた検討においては、いずれも単体の演算（乗算や除算）をいかに効率的に行うかに重点が置かれており、それらの演算を、特に組み合わせ回路を採用して数十～数百以上の演算を行うことについては、これまでほとんど検討がなされていないのが現状である。その理由は、種々推定することができるが、その1つとしては、復号演算は多くはこれまで順序機械によって実装され、組み合わせ回路を採用することは、得られる処理性能と回路規模の点で利点が少ないものと判断されていたこともある。

## 【 0 0 1 9 】

一方、誤り訂正のアルゴリズムについて考察すると、誤り位置の計算アルゴリズムにおいては、ガロア拡大体  $GF(2^m)$  上で定式化されたYule-Walker方程式がリード・ソロモン符号の計算において発生し、このYule-Walker方程式を効率よく処理することが高速化を達成し、かつ必要とされる回路サイズを可能な限り小さくするために必要とされる。Yule-Walker方程式を解くというアルゴリズムを実行させる場合、高速化の目的から組み合わせ回路で実現する場合には、誤り訂正能力の増大とともにYule-Walker方程式を解いて誤り位置を求める部分が組み合わせ回路の回路サイズの観点からみて非常に大きな割合を占めることになる。

## 【 0 0 2 0 】

加えて、リード・ソロモン符号の復号化を組み合わせ回路を用いて実現し、実際のシステムへと適用する場合には、処理に対する汎用性を付与し、付加的な回路または処理を追加しないためにも、任意の最小距離を持つリード・ソロモン符号の復号化に適用できるアルゴリズムを提供することが望ましい。特に光通信の分野ではITU(International Telecommunication Union)において(255、239) リード・ソロモン符号を使用することが標準化されたため、訂正可能な誤りの最大個数が8、最小距離が17である場合にも、リード・ソロモン符号の復号化を効率

的に行うことが可能なアルゴリズムが必要とされている。

【0021】

Yule-Walker方程式を解くという数学的な問題を、組み合わせ回路を用いてハードウェアとして実装可能な規模で実現するためには、回路規模の増大を抑制し、乗算器の個数を削減するアルゴリズム、およびこのアルゴリズムを効率よく処理する組み合わせ回路構成が必要とされている。すなわち、上述した誤り訂正装置および誤り訂正のアルゴリズムを、高速化といった目的を達成しつつ、許容可能な回路サイズとしてデバイス化を可能とする組み合わせ回路が必要とされていた。

【0022】

【発明が解決しようとする課題】

本発明は上述した従来の技術における不都合に鑑みてなされたものであり、本発明は、高速（40Gbpsもしくはそれ以上）の光通信分野、さらにより限定的には連続したデータを同期フレームとして転送するSONET上での波長多重通信のため、インターリーブされたリード・ソロモン符号の効率的な復号回路、復号装置、復号方法、および半導体デバイスを提供することにある。

【0023】

すなわち、本発明は、回路規模あたりの処理性能が高い（低レイテンシ・高スループット）、復号回路を提供することを目的とする。

【0024】

さらに、本発明の別の目的は、インターリーブされたリード・ソロモン符号にも上述した特長を失うことなく対応できる構成の、柔軟な復号回路を提示することにある。

【0025】

さらに本発明の別の目的は、インターリーブされた受信語のそれぞれの誤りパターン（誤り数、位置）に関係なく一定のサイクルで誤り語を連続して高速に出力する復号回路を提供することにある。

【0026】

さらに、本発明の別の目的は、ガロア拡大体 $GF(2^m)$ （ $m$ は2以上の任意

の自然数) 上の演算回路のうち、いくつかの入力が共通な複数の乗算 (例えば  $AB$  と  $AC$  と  $AD$ ) を行う回路、および積和演算  $AB + CD + EF + \dots$  を行う回路を、高速かつ高効率で処理でき、小規模な回路によって実現することが可能な組み合わせ回路、および上述した復号回路を含む復号装置、復号方法、および半導体デバイスを提供することにある。

## 【 0 0 2 7 】

さらに、本発明の別の目的は、リード・ソロモン符号の復号化を組み合わせ回路を用いて実現し、実際のシステムへと適用する場合に、処理に対する汎用性を付与し、付加的な回路または処理を追加せずに任意の最小距離を持つリード・ソロモン符号の復号化に適用できる復号回路、復号装置、復号方法、および半導体デバイスを提供することにある。

## 【 0 0 2 8 】

## 【課題を解決するための手段】

本発明の上記課題は、本発明の復号回路、復号装置、復号方法および半導体デバイスを提供することにより解決することができる。

## 【 0 0 2 9 】

本発明によれば、符号化された入力デジタル信号が符号のインターリーブ数に対応して並列して入力される入力部と、

前記入力部からのインターリーブされた符号語間でシリアル化されたデータ出力をそれぞれ処理する誤り位置多項式係数算出部と、誤り値多項式係数算出部とを含む処理部と、

前記処理部からの出力と前記入力デジタル信号とが入力され、前記処理部からの出力と、前記入力デジタル信号とから、誤りが訂正された出力デジタル信号をインターリーブ数に対応して並列に出力する出力部とを備える復号回路が提供できる。本発明の復号回路は、前記入力部は、前記入力デジタル信号のシンδροームを算出してインターリーブされた符号語間でシリアル化されたデータとしてシンδροームを前記処理部へと渡し、

前記処理部は、入力される前記シンδροームを使用して、誤り位置多項式係数と誤り値多項式係数とを算出し、

前記出力部は、前記処理部の出力を入力として誤り位置および誤り値を、前記入力デジタル信号からガロア拡大体上における線形演算から生成して前記出力デジタル信号とする。本発明の復号回路は、前記入力デジタル信号は、それぞれがリード・ソロモン符号化された  $i$  - チャンネルの並列入力とされ、さらに前記復号回路は、 $i : 1$  または  $1 : i$  のマルチプレクサまたはデマルチプレクサを含む。本発明の復号回路では、前記入力デジタル信号は、光通信における波長多重通信により送信される。本発明の復号回路の少なくとも前記入力部は、順序回路から構成され、前記処理部は、組み合わせ回路から構成される。本発明においては、デジタル信号の誤り訂正、または暗号化のために使用することができる。

### 【 0 0 3 0 】

本発明においては、符号化された入力デジタル信号を受信するための入力手段と、前記入力デジタル信号を処理して誤り位置多項式係数と誤り値多項式係数とを算出する処理手段と、前記誤り位置多項式係数と、前記誤り値多項式係数とから誤りが訂正された出力デジタル信号を出力する出力手段とを含み、

前記入力手段は、前記符号化された入力デジタル信号が符号のインターリーブ数に対応して並列に入力され、かつ前記符号化された入力デジタル信号からインターリーブされた符号語間でシリアル化されたデータとしてシンδροームを算出し、

前記処理手段は、前記受信手段から出力されるそれぞれの前記シンδροームを使用して誤り位置多項式係数と誤り値多項式係数とを算出し、

前記出力手段は、それぞれの前記誤り位置多項式係数と前記誤り値多項式係数と、前記符号化された入力デジタル信号とからガロア拡大体上における線形演算により誤りが訂正された出力デジタル信号を符号のインターリーブ数に対応して並列に出力する復号装置が提供される。本発明の復号装置においては、前記入力デジタル信号は、それぞれリード・ソロモン符号化された  $i$  - チャンネルの並列入力とされ、さらに前記復号装置は、 $i : 1$  または  $1 : i$  のマルチプレクサまたはデマルチプレクサを含む。本発明の復号装置では、前記入力デジタル信号は、光通信における波長多重通信により送信される。本発明の復号装置にお



いては、少なくとも前記入力部は、順序回路により構成され、前記処理部は、組み合わせ回路から構成される。本発明の復号装置は、デジタル信号の誤り訂正、または暗号化のために用いることができる。

### 【 0 0 3 1 】

本発明によれば、デジタル信号を復号するための方法であって、該方法は、符号化された入力デジタル信号を符号のインターリーブ数に対応して並列して入力するステップと、

前記入力部からのインターリーブされた符号語間でシリアル化されたデータ出力を誤り位置多項式係数算出部と、誤り値多項式係数算出部とを含む処理部を使用してそれぞれ処理するステップと、

前記処理部からの出力と前記入力デジタル信号とを入力し、前記処理部からの出力と、前記入力デジタル信号とから誤りが訂正された出力デジタル信号を生成するステップと、

前記生成された誤りが訂正された出力デジタル信号を、インターリーブ数に対応して並列に出力するステップと

を含む復号方法が提供される。本発明の復号方法においては、前記入力ステップは、前記入力デジタル信号のシンδροームを算出してインターリーブされた符号語間でシリアル化されたデータとしてシンδροームを前記処理部へと渡すステップを含み、

前記処理ステップは、入力される前記シンδροームを使用して、誤り位置多項式係数と誤り値多項式係数とを算出するステップを含み、

前記出力ステップは、前記処理部の出力を入力として誤り位置および誤り値を、前記入力デジタル信号からガロア拡大体上における線形演算から生成して前記出力デジタル信号とするステップを含む。本発明の復号方法においては、前記入力ステップは、それぞれがリード・ソロモン符号化された  $i$  - チャンネルの並列入力として前記入力デジタル信号を入力するステップを含み、さらに  $i : 1$  のマルチプレクシング・ステップおよび  $1 : i$  のデマルチプレクシング・ステップを含む。本発明の復号方法においては、光通信における波長多重通信により送信される入力デジタル信号として入力される。本発明の復号方法においては

、少なくとも前記入力ステップは、順序回路を使用した演算処理を行い、前記処理ステップは、組み合わせ回路を使用して演算処理を行う。本発明の復号方法は、デジタル信号の誤りを訂正するために使用することができる。

### 【 0 0 3 2 】

本発明によれば、デジタル信号を処理するために使用される半導体デバイスであって、該半導体デバイスは、

符号化された入力デジタル信号を受信するための入力手段と、

前記符号化された入力デジタル信号を処理して誤り位置多項式係数と誤り値多項式係数とを算出する処理手段と、

前記誤り位置多項式係数と、前記誤り値多項式係数とから誤りが訂正された出力デジタル信号を出力する出力手段とを含み、

前記入力手段は、インターリーブされた前記符号化された入力デジタル信号がインターリーブ数に対応して並列に入力され、かつ前記符号化された入力デジタル信号からインターリーブされた符号語間でシリアル化されたデータとしてシンδροームを算出し、

前記処理手段は、前記入力手段から出力されるそれぞれの前記シンδροームを使用して誤り位置多項式係数と誤り値多項式係数とを算出し、

前記出力手段は、それぞれの前記誤り位置と前記誤り値と、前記入力デジタル信号とからガロア拡大体上における演算により生成された出力デジタル信号をインターリーブ数に対応して並列に出力する、半導体デバイスが提供される。

本発明においては、前記入力デジタル信号は、それぞれリード・ソロモン符号化された  $i$  - チャンネルの並列入力とされ、さらに前記半導体デバイスは、 $i : 1$  または  $1 : i$  のマルチプレクサまたはデマルチプレクサを含む。本発明においては、前記符号化された入力デジタル信号は、光通信における波長多重通信により送信される。本発明においては、前記入力手段は、順序回路から構成され、前記処理手段は、組み合わせ回路から構成される。本発明においては、前記組み合わせ回路は、ガロア拡大体  $GF(2^m)$  ( $m$  は、2 以上の整数) におけるデジタル信号の 2 個以上の乗算を独立して行う複数の乗算器を含み、

前記乗算器は、入力側 XOR 演算部と、AND 演算部と、出力側 XOR 演算部

とを含んで構成され、前記入力側XOR演算部が、複数の前記乗算器に共有される。本発明においては、前記組み合わせ回路は、ガロア拡大体 $GF(2^m)$  ( $m$ は、2以上の整数)における積和演算器を含み、かつそれぞれの前記乗算器は、前記AND演算部と、前記出力側XOR演算部との間に接続される加算器を含み、前記出力側XOR演算部が共有されていて、複数の前記乗算器の前記AND演算部の出力を前記加算器により加算し、該加算結果を共有される前記出力側XOR演算部により演算する。本発明においては、複数の前記乗算器の入力が共通しており、前記入力側XOR演算部が複数の前記乗算器により共有される。

## 【0033】

本発明の半導体デバイスは、波長多重通信により送信されるデジタル信号の復号回路の誤り位置を算出する誤り位置算出部および誤り値算出部に使用される。本発明においては前記半導体デバイスは、復号または誤り訂正または暗号化に使用される。

## 【0034】

## 【発明の実施の形態】

以下本発明を、図面に示した実施の形態に基づいて説明するが、本発明は、図面に示した実施の形態に限定されるものではない。

## 【0035】

## セクション1 &lt;復号回路&gt;

図6には、光通信により送信されたデジタル信号に含まれる誤りを訂正するために使用することができる本発明の復号回路を示す。図6に示した復号回路は、入力部10と、処理部12と、出力部14とを含んで構成されている。入力部10は、例えば16インターリーブされた入力デジタル信号IDが入力される。処理部12は、この入力部からの出力を受け取って処理を行ない、誤り位置多項式係数と誤り値多項式係数とを算出する。出力部14は、この処理部12からの出力を受け取り、出力された誤り位置と誤り値とをAND処理し、入力される入力デジタル信号IDとXOR処理を行なって、出力デジタル信号ODを生成する。出力デジタル信号ODは、入力デジタル信号に含まれる可能性のある誤りが訂正されている。

## 【 0 0 3 6 】

図 6 に示した復号回路に入力される入力デジタル信号 I D としては、本発明においては、光通信、特に 4 0 G b p s といった高速のデータ転送レートで波長多重通信法により送信されるデジタル信号を入力することができる。より具体的には、上述した入力デジタル信号 I D は、例えば ( 2 5 5 , 2 3 9 ) R S 符号により符号長 2 0 4 0 ビットとして送信されるものを用いることができる。通常、波長多重通信方においては、上述した入力デジタル信号は、インターリーブ方式が採用されて、例えば 1 6 の並列な 2 5 5 バイト・ストリームとして本発明の復号回路に入力される。

## 【 0 0 3 7 】

図 6 に示した本発明の復号回路においては、上述した入力デジタル信号 I D は、入力部 1 0 においてインターリーブされて並列に入力され、それぞれの入力について、受信多項式が規定され、この受信多項式からシンドローム  $S_i$  が算出され、入力部 1 0 の出力とされる。入力部 1 0 (シンドローム算出部) の出力であるシンドローム  $S_i$  は、( 2 5 5 , 2 3 9 ) R S 符号の場合は、入力デジタル信号 2 5 5 バイトから得られる 1 6 バイトのデジタル情報として生成される。例えば、本発明の図 6 において説明する復号回路の実施の形態においては、入力デジタル信号は、2 0 4 0 ビットが 1 6 インターリーブされて 1 6 の 2 5 5 バイト・シリアル・ストリームとして入力部 1 0 へと入力され、入力部 1 0 において 1 6 のシリアル・ストリームに対応する 1 6 の 1 6 バイト・シンドロームが生成されることになる。

## 【 0 0 3 8 】

図 6 に示されるように、入力部 1 0 においては、それぞれの入力デジタル信号 I D のシリアル・ストリーム I D S に対して 1 つのシンドローム算出部 1 6 が割り当てられ、シンドロームが算出される構成とされている。算出されたそれぞれのシリアル・ストリーム I D S についてのシンドロームは、レジスタ 1 8 に保持された後、出力部 1 4 へとマルチプレクサにより符号語間でシリアル化されたデータとして出力される。例えば、上述したように、1 6 インターリーブされた 2 5 5 バイトの入力デジタル信号から得られた 1 6 バイトのシンドロームから

は、誤り位置および誤り値の算出の為に、128ビットの信号が得られることになる。図6に示した入力部10において使用することができるシンドローム算出部16としては、順路回路を使用する回路など、これまで知られたいかなる回路でも使用することができる。なお、シンドロームの定義および算出方法についてはより詳細に後述する。

## 【0039】

図6に示された復号回路においては、算出されたそれぞれのシンドローム $S_i$ は、処理部12へとそれぞれ順次入力される。図6においては説明の便宜のため、処理部12は、1つだけを例示して示している。図6に示した処理部12は、複数の乗算器により構成された組み合わせ回路から構成される、誤り位置多項式 $\Lambda(x)$ を算出するための誤り位置多項式係数算出部18と、誤り値多項式 $E_r(x)$ を算出するための誤り値多項式係数算出部20と、を含んで構成されている。処理部12において使用される乗算器を含む組み合わせ回路の詳細についてはセクション2<組み合わせ回路>においてより詳細に説明する。

## 【0040】

図6に示した処理部12の出力である誤り位置多項式 $\Lambda(x)$ と、誤り値多項式 $E_r(x)$ とは、出力部14へと、図示しないデマルチプレクサにより、例えばインターリーブの数に対応する用にデマルチプレクスされた後、入力される。出力部14には、それぞれの入力デジタル信号のインターリーブの数に対応する数だけ配置されたレジスタ22と、ANDゲート24と、XORゲート26とが形成されている。出力部16では、それぞれのシンドロームから得られた誤り位置情報（誤り位置には「1」、そうでなければ「0」） $\Lambda_{eval}$ を使って、誤り値 $E_r$ をANDゲート24により選択する。さらに選択された出力は、XORゲート26において加算が行われる。このXOR選択ゲート26には、符号化された入力デジタル信号IDから得られるシリアル・ストリームIDSが、 $imn$ ビットのバッファ28a、28bを介してXORゲート26へと入力されており、XORゲート26においてガロア拡大体 $GF(2^m)$ 上の減算を実行させることによって誤りが除去された255バイトの出力デジタル信号ODとして与える構成とされている。

## 【 0 0 4 1 】

図 6 に示した本発明の復号回路においては、上述した処理部 1 2 を構成する組み合わせ回路を順序回路として構成することもできるが、本発明において特に複数の乗算器を、入力側 X O R 演算群（変数前処理部）と、A N D 演算群、出力側 X O R 演算群（剰余演算）の 3 段構成とし、変数前処理部または剰余演算部の一方ないし両方を、複数の乗算器間で共有させる構成を採用することにより、従来誤り位置および誤り値の算出においてクリティカル・ポイントとされてきた処理部 1 2 における回路規模を実用上許容可能な規模としつつ、効率的に乗算器から構成させることが可能となる。

## 【 0 0 4 2 】

図 6 に示した本発明の復号回路では、上述したように出力部 1 4 を、処理速度を低下させる非線形演算を実行させる回路構成を採用することなく、定数乗算器と加算器といった線形演算を行う回路のみで構成させる。このため、本発明の復号回路は、処理速度を低下させず高速に、かつ回路構成を小規模としつつ復号回路として構成することが可能となる。さらに本発明者らは、鋭意検討の結果、特定の構成を含む乗算器から構成される組み合わせ回路と、該組み合わせ回路において誤り位置および誤り数を効率的に算出することを可能とするアルゴリズムとを採用することにより、従来にまして柔軟で高速かつ、小規模の復号回路を構成することを見出し、本発明に至ったものである。

## 【 0 0 4 3 】

本発明において採用する誤り位置多項式係数算出の方法またはアルゴリズムについてはより詳細に組み合わせ回路構成と共に後述するが、以下に、本発明の復号回路における処理部 1 2 に含まれる誤り値多項式係数算出部 2 0 の機能・作用について説明する。

## 【 0 0 4 4 】

## 誤り値計算アルゴリズムの選択

本発明の復号アルゴリズムについては、誤り位置の評価だけではなく誤り値の評価も  $O(t)$  次の多項式計算（線形演算）で直接計算可能なアルゴリズムをインターリーブされたリード・ソロモン符号の復号に対して適用するものである。

この際、本発明において採用するアルゴリズムにおいては、特に誤り値の計算に必要な除算を、多項式の評価の後の出力のクリティカル・パス中で誤り位置ごとに行うのではなく、多項式の評価の前に符号語ごとにただ一度線形演算で行う。これにより、多項式評価の値を誤りの値として一定のサイクルで直接高速出力することが可能となる。

【0045】

さらに、誤り値  $E_r(x)$  多項式の次数は、 $t$  個の独立した出力を出すのに最低必要とされる  $t-1$  次まで低くできることが判明している。その際に得られる  $t$  個の係数は、誤り位置多項式の係数とシンδροームとを使って計算することが可能である。このアルゴリズムの採用により、シンδροーム計算、ならびに、誤り位置評価だけではなく、誤り値の評価も線形演算回路だけで実行することが可能となり、入出力出力回路全体を簡略化高速化することが可能となる。

【0046】

本発明の復号アルゴリズムまたは復号方法において種々の誤り値多項式を使用することができるが、以下、 $e$  個の誤りが  $i_0, \dots, i_{e-1}$  に発生したときの誤り値多項式  $E_r(x)$  が下記式

$$E_r^{(e)}(x) = \sum_{l=0}^{e-1} \frac{E_{i_l} \prod_{j \neq l} (x + a^{i_j}) \prod_{j, k \neq l, j < k} (a^{i_j} + a^{i_k})}{\prod_{j < k} (a^{i_j} + a^{i_k})}$$

の形式で与えられ、除算を含むが、分母は多項式ではなく、符号語ごとに定数である実施の形態を例として、本発明の機能・作用について説明する（上記式中、 $a$  は、ガロア拡大体の原子元を意味する）。

【0047】

上述した誤り値多項式  $E_r^{(e)}(x)$  は、多項式  $E_r^{(e)}(x)$  の計算に位置

$$a^{i_k}$$

（以下  $a^{i_k}$  と略する）に存在する  $k$  番目の誤り値

$$E_{i_k}$$

(以下  $E_{ik}$  と略記する) が必要となるので本末転倒であり、直接本発明の復号回路に使用することができない。しかしながら、もし  $E_r^{(e)}(x)$  中のすべての  $E_{ik}$  といくつかの  $a^{ik}$  を、シンδροーム  $S_i$  を使って記述すれば、回路化が可能となる。さらに、この場合  $E_r^{(e)}(x)$  中のすべての  $a^{ik}$  を誤り多項式  $\Lambda^{(e)}_j$  を使用して記述することで、誤り位置を求める前に誤り値の算出を実行することが可能となる。したがって誤り値の算出を並列化することが可能となり、この結果高速化を達成することができる。

【0048】

上述したプロセスは、以下の復号アルゴリズムまたは復号方法を使用することにより実行することができる。まず、分母の部分を  $\Lambda^{(e)}_j$  で記述する。誤り値と誤り位置多項式の係数は、定数ファクタを除いて誤り位置、

$$a^{i_0}, a^{i_1}, a^{i_2}, \dots, a^{i_{t-1}}$$

について、それぞれ Elementary symmetric function となる。例えば、誤り位置多項式の係数

$$\Lambda_1, \Lambda_2, \dots, \Lambda_t$$

については、

$$\begin{aligned} \Lambda_1 &= \sum_k a^{i_k} \\ \Lambda_2 &= \sum_{k < l} a^{i_k} a^{i_l} \end{aligned}$$

:

$$\Lambda_{t-1} = a^{i_0} a^{i_1} a^{i_2} \dots a^{i_{t-1}}$$

となっており、誤り位置、

$$a^{i_0}, a^{i_1}, a^{i_2}, \dots, a^{i_{t-1}}$$

は、互いに交換可能である。また、Shur関数と呼ばれる2つのVandermonde行列式の割り算で定義されるものと、上述した Elementary symmetric function との



間に成り立つ関係式(例えば、I.G.Macdonald, Symmetric Functions and Orthogonal Polynomials, American Mathematical Society, 1998参照)と、ガロア拡大体  $GF(2^m)$  において成り立つ加算と2乗算の下記交換関係、

$$(a+b)^2 = (a-b)^2 = a^2 + b^2$$

から、以下の新規な関係が導出できる。

$$f^{(e)} = \prod_{m>l} (a^{i_m} + a^{i_l})$$

$$= \begin{vmatrix} \Lambda_1^{(e)} \Lambda_3^{(e)} & \cdots 0 & \cdots & 0 \\ 1 & \Lambda_2^{(e)} \Lambda_4^{(e)} & \cdots & 0 & \cdots 0 \\ 0 & \Lambda_1^{(e)} \Lambda_3^{(e)} & \cdots & 0 & \cdots 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & \cdots \Lambda_{e-2}^{(e)} \Lambda_e^{(e)} \\ 0 & \cdots & 0 & \cdots \Lambda_{e-3}^{(e)} \Lambda_{e-1}^{(e)} \end{vmatrix}.$$

【0049】

上記行列式を採用することにより、さきの  $Er^{(e)}(x)$  の分母を、 $\Lambda^{(e)}$  で記述することが可能となる。 $Er^{(e)}(x)$  の分子についても同様に算出してみると、下記式で示すように、 $Er^{(e)}(x)$  の係数が  $S_i$ 、 $\Lambda^{(e)}_i$  のみで記述でき、 $E_{ik}$  と、 $a^{ik}$  を使用しなくとも算出することができることになる。

$$Er^{(e)}(x)$$

$$= \frac{\sum_{k=0}^{e-1} E_{i_k} \prod_{j \neq k} (x + a^{i_j}) \prod_{m, l \neq k, m > l} (a^{i_m} + a^{i_l})}{\prod_{m > l} (a^{i_m} + a^{i_l})}$$

$$= \frac{\sum_{k=0}^{e-1} E_{i_k} \left( \sum_{j=0}^{e-1} x^j \Lambda_{e-j-1, i_k}^{(e)} \right) f^{(e-1)}(\Lambda_{1, i_k}^{(e)}, \Lambda_{2, i_k}^{(e)}, \dots, \Lambda_{e-1, i_k}^{(e)})}{f^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \dots, \Lambda_e^{(e)})}$$

$$= \frac{\sum_{k, j, m=0}^{e-1} E_{i_k} a^{i_k m} x^j Er_{jm}^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \dots, \Lambda_e^{(e)})}{f^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \dots, \Lambda_e^{(e)})}$$

$$= \frac{\sum_{j, m=0}^{e-1} S_m x^j Er_{jm}^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \dots, \Lambda_e^{(e)})}{f^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \dots, \Lambda_e^{(e)})}.$$

ここで、

$$\Lambda_{jik}^{(e)} = \Lambda_j^{(e)} + a^{ik} \Lambda_{j-1}^{(e)}$$

であり、具体的には、 $a^{ik}$ を除いた誤り位置に対応する誤り位置多項式の係数である。

## 【 0 0 5 0 】

上述したように、上述した関係を新たに採用することにより、非線形演算回路を使用することなく、ガロア拡大体  $GF(2^m)$  上における線形演算回路により、処理部 14 を構成することが可能となり、高速化を実現することが可能となる。

## 【 0 0 5 1 】

以下、上述の誤り値計算アルゴリズムを用いてインターリーブ符号にも対応した高速、小回路規模の符号回路を構成する為に以下の構成を採用する。

(1) 高速の入力部・出力部（線形演算回路）の採用

まず、符号の構成（インターリーブ数）と入出力インターフェースのバス幅、クロック数等を考慮して、入力に接続されたシンドローム計算のための多項式評価（線形演算）回路、出力側に接続された誤り位置評価・誤り値評価のための多項式評価（線形演算）回路を、RS符号が巡回符号であることに起因するサイクリックな構造を生かして、1からnの任意のクロック数で(n,k)リードソロモン符号の前処理・後処理する高速な順序回路として実現したことである。特に、従来方式でクリティカル・パスであった誤り値評価の部分に対しても上述した構成を採用するこちが本発明においては有効である。また、本発明の構成によれば、符号語あたりの入力されるデジタル信号幅が255バイトの場合ばかりではなく、1、3、5、15、17、51、85バイトインターフェースとして、柔軟に対応することができる復号回路を提供することができる。

## 【 0 0 5 2 】

表1には、本発明の復号回路において入力デジタル信号の入力幅に対して要求される処理クロックの関係を示す。

## 【 0 0 5 3 】

【表 1】

表 1 : 符号語あたりの入出力幅とクロック数 (n=255)

入出力幅 (バイト)	1	3	5	15	17	51	85	255
符号語 (n = 255) 処理に必 要なクロック数	255	85	51	17	15	5	3	1

表 1 に示されるように、デジタル信号の入力幅が小さくなればそれに対応して必要なクロック数は増加するものの、本発明の復号回路を採用することにより柔軟に対応することが可能となることが示されている。

## 【 0 0 5 4 】

さらに、本発明においては、符号語あたりの入出力のバイト幅は表 1 で示された以外でも任意に選択可能である。例えば入出力幅 8 バイトの場合でも、符号の長さを最後にダミー 1 バイトを加えて n=256 バイトとすることにより対応可能となる。

## 【 0 0 5 5 】

## (2) 非線型演算回路 1 2 との接続

本発明においてはさらに、入力部 1 0 および出力部 1 4 として順序回路を用いて構成し符号のインターリーブ数に応じて複数用意し、前後の順序回路の間にシンδροームならびに誤り位置・誤り値多項式の係数を保持する回路と、マルチプレクサ・デマルチプレクサとを用いて、誤り位置・誤り値多項式の係数計算を行う非線型演算回路 1 2 に接続する。この演算回路は上述したように、非線形演算を実行する乗算器といった非線形演算回路の組み合わせ回路として構成されている。このため、本発明においては、例えば、OC-768 を仮定して (255, 239) リードソロモン符号を 1 6 インターリーブして使用する場合には、前後に 1 6 個ずつ順序回路を用意することが必要となる。すなわちこの場合、1 6 : 1 のマルチプレクス・デマルチプレクスを行う必要がある。しかしながら、本発明においては、シンδροーム多項式の係数をマルチプレクス・デマルチプレクスするために、2040 ビットではなく、1 2 8 ビット ( $E_r(x)$ ) の取り方によっては、後段は、136 ビット) の信号を扱うだけですみ、マルチプレクサ・デマルチプレクサ、バッフ

アの数共に大きく削減することが可能となる。

【0056】

(3) 中央の非線型演算回路を時分割に使う3段パイプライン動作方式

本発明においては、さらに、復号回路全体を線形演算回路（シンδροーム計算）－非線型演算回路（誤り位置・誤り値多項式の係数計算）－線形演算回路（誤り位置・誤り値の評価）の3段パイプラインとして動作させ、低レイテンシの非線型演算回路をインターリーブされたそれぞれの符号の計算にシンδροームを符号語間でシリアルに順次与える事により時分割して用いる。このため、回路規模あたりの処理能力の高い高効率のインターリーブ符号の復号動作を実現できる。例えば、OC-768のケースでは、中央の非線型演算回路は、上述した3段パイプラインの動作をさせるためには、約40nsのレイテンシでインターリーブされたそれぞれの符号語あたりの処理を終える必要があるが、最先端の半導体技術（0.18 $\mu$ m以上）を使い、組み合わせ回路を用いた実装を行うことにより、上述した復号回路をASICといった半導体バイスとして実現できる。

【0057】

すなわち、本発明の復号回路は、(1) 高速の入力部10と高速の出力部14を採用すること、特に従来では非線形演算を行う必要があった出力部14に対して、誤り位置および誤り値を線形演算回路だけを使用して算出できるようにし、さらに、処理部12に対しては上述した復号アルゴリズムに適合すると共に、回路サイズを低減させる乗算器構成を採用すること、(2) 本質的に非線形演算を実行する処理部12を、時分割して使用して、3段パイプライン動作方式を採用すること、による相乗的な効果により、高速かつ回路サイズが許容範囲の復号回路、誤り訂正装置を提供することを可能とするものである。以下、セクション2として、本発明の復号回路における処理部12に含まれる乗算器から構成される組み合わせ回路について詳細に説明する。

【0058】

セクション2＜組み合わせ回路＞

本発明の復号回路に使用する処理部12は、線形演算回路、具体的には乗算器を使用した組み合わせ回路として構成される。本発明において使用される乗算器

は、しかしながら、従来の組み合わせ回路に使用される乗算器は、ガロア拡大体  $GF(2^m)$  における乗算をAND演算群の次にXOR演算群を行う2段階の構成とするのではなく、XORゲート-ANDゲート-XORゲートの3段階の構成を採用する。

## 【0059】

## 単体の並列乗算回路の構成方式

単体の乗算回路に関しては従来より多くの検討がなされているものの、順序回路でなく組み合わせ回路として構成された並列乗算回路 (Mastrovito Multiplier) については意外に研究の歴史が浅く、近年検討が開始されたといってもよい。従来の並列乗算回路 (以下、本明細書においては、単に乗算回路という。) の構成方式は、AND-XOR形式とXOR-AND-XOR形式の2種類で、相互変換可能である。ただし単体の乗算しか回路化しない場合には、AND-XOR形式のものが通常用いられる。その理由は、AND-XOR形式はよく研究されていて小規模回路を得る方法がかなり検討されているのに対し、XOR-AND-XOR形式については一般に回路規模が削減される保証がなく (むしろ増える場合がある)、設計作業の複雑化に見合うだけの削減効果がないと考えられることが、その理由であると考えられる。以下、各場合について検討を加える。

## 【0060】

## (1) AND-XOR形式

この方法は、筆算どおりに計算を進める教科書的な方法で、通常はこの形の回路を使用する。具体的には、乗算の引数となる2つの  $(m-1)$  次多項式に対し、その係数どうしを組み合わせ  $m^2$  個の部分積をまず作る。これがAND部の処理内容である。次に、それらの部分積のうち次数が同じものどうしを加算して  $(2m-2)$  次多項式を構成し、既約多項式による剰余演算を行って  $(m-1)$  次の解を得る。これらがXOR部の処理内容である。ANDの個数は  $m^2$ 、XORの個数は  $O(m^3)$  であるが、既約多項式や基底を選択することによりXORが  $(m^2-1)$  個で構成できることが広く知られている。任意の乗算回路は必ずこの形式で構成できる。

## 【0061】

## (2) XOR-AND-XOR形式

上述したAND-XOR形式の回路に対し、ブール代数の規則である  $(A \text{ and } B) \text{ xor } (A \text{ and } C) = A \text{ and } (B \text{ xor } C)$  のもとで、剰余演算部にあるXOR演算をANDの前に移動して変数前処理演算部（入力側XOR演算）とすることが、一般に可能である。これによって乗算器の回路を、XOR-AND-XOR形式とすることができる。XORの移動にあたって、 $A \text{ xor } A = 0$ 、 $B \text{ xor } 0 = B$ の性質を利用し、剰余演算部（出力側XOR演算）のXOR中に同一の冗長タームを偶数個追加しておくことで、より多くのXOR演算を変数前処理部へ移動できる場合がある。この操作が可能のため、XORの移動には、単に分配律をそのまま適用するのみならず多様なやり方が存在し得る。したがって、たとえ同一の基底や既約多項式のもとであっても、XOR-AND-XOR形式は複数存在することになる。ゲート数は、XOR-AND-XOR形式にすることでAND-XOR形式より増加する場合も減少する場合もあり、まちまちである。また、次に述べるComposite Field Multiplierのように、特殊な基底のもとでシステマティックにXORゲートを削減する方法もこれまでに知られている。

【0062】

## (3) 限定された体のみに適用可能なXOR-AND-XOR形式の構成法 (Composite Field Multiplier)

Composite Field Multiplierは、 $m$ が合成数で、なおかつ体の要素の表現に用いる基底が通常基底（多項式基底や正規基底）でなくてもよいという、特殊な場合に限って使える乗算回路構成法である。以下、より詳細に説明を行う。 $m$ が合成数のとき、 $GF(2)$ から2回以上の体の拡大によって拡大体 $GF(2^m)$ を構成できる。その拡大の過程に従って再帰的な構造の乗算回路を構成するのがComposite Field Multiplierである。このとき、例えば1回の2次拡大に対し、 $GF(2^m)$ 上の2つの値 $Ax+B$ と $Cx+D$ （ここで $A, B, C, D$ は、それぞれ部分体 $GF(2^{m/2})$ 上の値）の積が、

$$(Ax+B)(Cx+D) = ACx^2 + ((A+B)(C+D) + AC + BD)x + BD$$

であることを用いれば、部分体上の乗算を4個から3個へ減らすことができ、回路規模を減少させることができる(KOA)。同時に、回路はXOR-AND-XORの構造と

して構成することができる（乗算の前に行う加算が、ANDの前に配置されるXORに対応する）。なお、KOAの使用はComposite Field Multiplierであることが前提であって、そうでない乗算回路ではKOAは通常使用することができない。また、仮にmが同手法を適用できる値であっても、体の変換回路が必要でそのオーバーヘッドのためにかえって回路規模が増えるので、単体の乗算しか回路化しない場合には同手法は通常採用される構成とはならないものである。

## 【 0 0 6 3 】

＜本発明における組み合わせ回路の通常の乗算器による構成＞

本発明が対象とする入力共通乗算回路群、および積和演算回路を通常のAND-XOR構成で構成した場合の構成を図7および図8に示す。図7においては組み合わせ回路の例として、2つの乗算器を使用する組み合わせ回路が示されている。図7に示されるように従来の乗算器は、第1の入力A1が乗算器40および乗算器42へと入力され、乗算器40には、第2の入力B1が入力されて第1の出力45が、乗算器42には第3の入力B2が入力されて第2の出力46とが出力されている。図7に示すとおり、従来の構成では、乗算間で共通な入力があっても、そのために共有可能となる回路はまったく存在しない。図8には、積和演算を行うための組み合わせ回路を従来の乗算器の構成により構成した従来例を示す。図8に示す積和演算を行うための組み合わせ回路においても、そのままでは共有可能な回路は存在しないことがわかる。

## 【 0 0 6 4 】

図9は、従来の構成の乗算器を使用する組み合わせ回路の例を示す。図9の入出力では、1シンボルを1本の線として表記してある。また、入出力はそれぞれ8ビット幅であるものと仮定している。図9における組み合わせ回路においては、6シンボルの入力と1シンボルの出力とを含み、さらに7個の乗算回路と5個の加算回路を含んで構成されている。図9に示した組み合わせ回路においては、 $S_0, \dots, S_3Q$ で示された入力が乗算器群46へと入力され、加算器群48により加算された後、積和演算回路50へと入力されて、各入力の積和演算の結果である出力 $L_{21}Q$ が生成されている。

## 【 0 0 6 5 】

図9においては、破線で示したクロスターム構成演算と剰余演算の組み合わせが1つの乗算に相当する。図9において示されている乗算器の回路は標準的なものなので、詳細な回路の説明については省略する。この乗算回路を含む組み合わせ回路は、64AND+約103XORの数のゲートを含んでおり、回路全体のゲート数は448AND+約761XORとなる。図9から明らかのように、ほとんどの乗算の入力は、その一方あるいは両方が他の乗算と共通である。また、最終段などで積和演算が行われている。

## 【0066】

表2には、ガロア拡大体 $GF(2^8)$ において従来の標準的なAND-XORの2段構成の乗算回路、Composite Field Multiplierおよび部分体拡大体 $GF(2^4)$ の乗算をXOR-AND-XORにする改変を行った乗算回路に含まれる各ゲートの数を示す。

## 【0067】

【表2】

	変数前処理		クロスターム構成		剰余演算
標準的な AND-XOR の乗算回路	なし	+	64AND	+	103XOR
Composite Field Multiplier	4 XOR*2 個	+	48AND	+	56XOR
本発明の効果を挙げるために、部分体拡大体 $GF(2^4)$ の乗算を XOR-AND-XOR にしたもの					
a. 3個中3個の部分体乗算を改変	2 2 XOR*2 個	+	30AND	+	44XOR
b. 3個中2個を改変	1 6 XOR*2 個	+	36AND	+	48XOR
c. 3個中1個を改変	10XOR*2 個	+	42AND	+	52XOR

## 【0068】

すなわち、単体の乗算器のみに着目すれば、上記a~cいずれも回路規模は従来のComposite Field Multiplierの場合より増加しており、最小規模のものではない。このように、単に乗算器をXOR-AND-XORの3段構成とするのでは、回路の規模を逆に大きくする場合もある。

## 【0069】

＜本発明の組み合わせ回路における乗算器構成＞

一般には、多数の乗算や積和演算が絡み合うとブール代数上での最適化は困難



である。しかしながら、本発明における処理部 1 2 として組み合わせ回路を使用することを考慮すれば、上述した積和演算や乗算が並列かつ多段に多数接続された構造となっており、 $i$  段目の積和演算は一般に 0 段目～ $i - 1$  段目の出力を入力とする。したがって、後段の演算回路になるとほぼ組み合わせ回路の回路全体において、共通した入力を並列処理する必要が生じるため、最適化範囲が広がることになる。本発明者らは、この点に着目し、他の演算とのバランスも取りつつ、ブール代数上の最適化することにより、乗算器の効率的な組織化を達成したものである。

## 【 0 0 7 0 】

図 1 0 は、図 7 に示した従来の構成の乗算器および加算器からなる組み合わせ回路を、乗算器を 3 段構成として本発明の組み合わせ回路とした本発明の実施の形態の組み合わせ回路を示す。図 1 0 に示される組み合わせ回路においては、入力 A 1 が第 1 の XOR 群 5 2 へと入力され、入力 B 1 が第 2 の XOR 群 5 4 へと入力され、入力 B 2 が、第 3 の XOR 群 5 6 へと入力される構成とされている。これらの XOR 群 5 2、5 4、5 6 が、本発明において採用される変数前処理演算を実行するゲートである。このうち XOR 群 5 2、5 4、5 6 は、共通した入力 A 1 を処理する構成とされていて、回路規模の縮少になっている。各 XOR 群 5 2、5 4、5 6 の出力は、それぞれ AND 群 5 8 へと入力され、クロスタームが算出された後、再度下流側の XOR 群 6 0 において剰余演算が実行され、XOR 群 6 0 a から出力 6 2 が出力され、XOR 群 6 0 b から出力 6 4 が生成されている。図 1 0 にいては、1 つの乗算を行う単位が破線 B L で示されており、変数前処理 (XOR) - クロスターム演算部 (AND) - 剰余演算部 (XOR) の 3 段構成により、1 つの乗算器が構成されているのが示されている。図 1 0 に示されるように、各乗算を XOR-AND-XOR の構成とし、かつ、共通の入力に対して行う XOR 演算を乗算間で統一すれば、その XOR 演算回路を乗算回路間で共有できることになる。1 個分の乗算に相当する部分 (変数前処理演算 \* 2 + 新しいクロスターム構成部 + 新しい剰余演算部) の回路規模が通常の乗算回路と同じか若干大きい程度であれば、乗算群全体としての回路規模を減少させることが可能となる。

## 【 0 0 7 1 】

図 1 1 は、図 1 0 に示した従来の組み合わせ回路を、本発明において採用する 3 段構成の乗算器を使用して実装化した場合の組み合わせ回路の構成の別の実施の形態を示した図である。図 1 1 に示される組み合わせ回路においては、入力 6 6、入力 6 8、入力 7 0、入力 7 2 が、それぞれ変数前処理演算を実行する XOR 群 7 4、7 6、7 7、7 8 へと入力され、XOR 群のそれぞれの出力が、クロスターム構成演算を実行する AND 群 8 0、8 2 へと入力されている。

## 【 0 0 7 2 】

AND 群 8 0、8 2 の出力は、加算回路 8 4 へと入力されて加算が実行され再度共有された下流側の XOR 群 8 6 において剰余演算が実行されて、乗算が行われ、出力 8 8 が生成される構成とされている。図 1 1 に示した乗算器 1 単位の構成は、枠 B x で示した内側において形成されている。図 1 0 との相違点は、入力側の XOR 群 7 4、7 6、7 7、7 8 が共有されない場合であっても、本発明においては、下流側、すなわち出力側の剰余演算を実行する XOR 群 8 6 を共有する構成とすることができる。

## 【 0 0 7 3 】

さらに、本発明においては、入力側、すなわち変数前処理演算を実行する XOR 群 7 4 ~ 7 8 を共有化し、剰余演算を実行する XOR 群を同時に共有させることにより、さらに全体としての組み合わせ回路の構成を縮小することができる。

## 【 0 0 7 4 】

図 1 2 は、図 9 に示した組み合わせ回路を、3 段構成の乗算器を使用して本発明の組み合わせ回路とした、さらに別の実施の形態を示した図である。なお、図 1 2 に示した組み合わせ回路は、図 6 において示した RS 符号誤り訂正のための復号回路 ( $e = 2$ ) の処理部 1 2 の一部を構成する組み合わせ回路の実施の形態である。図 1 2 に示した組み合わせ回路においても、図 9 に示した従来例と同様に  $S_0 \sim S_3 Q$  が入力されている。図 1 2 に示すように、入力  $S_0$  の変数前処理演算を行う XOR ゲート 9 0 は、枠で示すように、AND ゲート 9 2、9 4、9 6 に対応する 3 つの乗算器により共有されている。さらに下流側においては、剰余演算部 9 8 についても複数の乗算器により共有されていて、変数前処理演算を

行うXORゲートおよび剰余演算を実行するXORゲートの双方が共有されているのが示されている。また、図10、図11、図12と同様に、乗算器の1つの単位は、鎖線により示されている。図12に示した組み合わせ回路においては、回路中の変数前処理演算部は8個、クロスターム構成演算部は7個、剰余演算部は4個、8ビット幅の加算は2個、クロスターム構成演算部と同ビット幅の加算は3個となる。

## 【0075】

このことから、表2で示した乗算回路をもとに、図12に示した組み合わせ回路で上述した復号回路の一部を構成させる場合については、そのゲート数は、以下の表3のようにまとめることができる。

## 【0076】

## 【表3】

Composite Field Multiplier	416XOR + 336AND 計 752gate
本発明	
a. 3個中3個の部分体乗算を改変	458XOR + 210AND 計 668gate
b. 3個中2個を改変	444XOR + 252AND 計 696gate
c. 3個中1個を改変	430XOR + 294AND 計 724gate

## 【0077】

表3 a～cに示されるように、単体の乗算をあえて最小規模にはしなかったにもかかわらず、回路全体としてはより回路規模が縮小されることが見出された。

## 【0078】

図13は、誤り訂正能力  $t = 2 \sim 8$  の図6に示した誤り訂正回路の処理部12に使用する場合において必要とされるXORゲート数と、それに対応する乗算器数とを示した図である。図13は、縦軸をXORの全数とし、横軸を乗算器の個数として示されており、この場合には、図6の誤り訂正用の復号回路は、 $m = 8$ で、既約多項式が  $x^8 + x^4 + x^3 + x^2 + 1$  であるものとして算出した。この場合、乗算を662、加算を531、2乗演算を30回使用する。表3および図13から理解されるように、乗算器単体の乗算をあえて最小規模にはしなかった

にもかかわらず、本発明の構成を採用することにより、組み合わせ回路全体としては回路規模をより縮小することができることが示される。

## 【 0 0 7 9 】

図 1 3 に示した実施の形態においては、具体的な回路構造を明確にする目的から、乗算における変数前処理部、クロスターム構成部、剰余演算部をすべての乗算について同一であるものとして説明した。実装上の段階においては、さらに良好な結果を得るために、どれだけのXORを変数前処理部・剰余演算部に配置するか（つまり、表記a～cのいずれを用いるか）を、回路中の乗算ごとに変更して最適化することも可能である。さらに、図 1 3 に示した実施の形態においては、回路入力に対する演算数の割合が少なく、体変換のオーバーヘッドもあるので、ゲート数減少の効果はさほど大きくはないものの、実用上は入力に対する演算の割合がかなり高くなるため、ゲート数削減効果による回路規模の削減は、図 1 4 に示すように顕著なものとなる。

## 【 0 0 8 0 】

## セクション 3 &lt;誤り訂正アルゴリズム&gt;

以下、本発明の復号回路、誤り訂正装置において使用される誤り訂正アルゴリズムについて詳細に説明する。

## 【 0 0 8 1 】

（従来技術の概要）

A. <Yule-Walker方程式を解く、または誤り位置多項式を求めるための従来の手法とその問題点>

本発明においては、 $GF(2^m)$ 上で定義された次の連立一次方程式の解を組み合わせ回路を用いて計算するための効率的なアルゴリズムを見出すことが必要である。

## 【 0 0 8 2 】

$$\begin{pmatrix} S_0 & S_1 & \cdots & S_{l-1} \\ S_1 & S_2 & \cdots & S_l \\ \vdots & & \ddots & \vdots \\ S_{l-1} & S_l & \cdots & S_{2l-2} \end{pmatrix} \begin{pmatrix} \Lambda_l^{(l)} \\ \vdots \\ \vdots \\ \Lambda_1^{(l)} \end{pmatrix} = \begin{pmatrix} S_l \\ \vdots \\ \vdots \\ S_{2l-1} \end{pmatrix}$$

上記式中、

$$S_0, S_1, \dots, S_{2t-1}$$

は、与えられた  $GF(2^m)$  の元であり、 $\Lambda^{(1)}_i$  が未知の量である。

【0083】

上記の連立一次方程式において、左辺の行列は、右斜め方向(対角線と交わる方向)に同じ成分が並ぶという規則的な構造をしており、Hankel行列といわれる。一般にこのタイプの方程式は、Yule-Walker方程式と呼ばれ、誤り訂正符号の理論をはじめとして時系列解析や信号処理の分野でも現れるなど、広い応用範囲を有していることが知られている。誤り訂正アルゴリズムにおいては、誤り位置多項式を決定する部分にこのYule-Walker方程式が現れることになる。そこで、本発明においては、上述したYule-Walker方程式の解を得るためのアルゴリズムを、リード・ソロモン符号の復号を行うために使用される誤り訂正アルゴリズムに適用するものである。

【0084】

上述したYule-Walker方程式の解法としてよく知られているものとして、Levinsonのアルゴリズム、Levinson-Durbinのアルゴリズム等がある。これらのアルゴリズムは、いずれも行列のサイズ1が小さい所から計算を始め、再帰的に行列のサイズが大きい方程式の解を決定していくものである。また、計算量は共に  $1^2$  のオーダーである。しかしながら、これらのアルゴリズムは、計算ステップの中に割り算の操作を含んでいる。このことは、アルゴリズムの実行を組み合わせ回路として実装することを考えたとき、分母が0か否かに応じた条件分岐が発生することを意味する。この条件分岐が発生することにより、条件分岐の各々に対して別々の回路を用意しなくてはならないので必要とされる回路サイズは 行列のサイズが大きくなるにしたがって組み合わせ的な速さで大きくなってしまいうという本質的な問題を生じる。

【0085】

また、本発明においては、Yule-Walker方程式の解を求めることによって、特にリード・ソロモン符号の復号化において、誤り位置多項式を決定することを目

的とする。従来Yule-Walker方程式の解法として用いられている方法としては、Peterson法、Berlekamp-Massey法、Euclid法などを挙げることができる。これらはいずれも訂正可能な誤りの最大数 $t$ に関して、多項式オーダの計算量で誤り位置多項式の係数を計算するものである。しかしながら、Berlekamp-Massey法とEuclid法を組み合わせた回路で表現と以下に述べる問題が生じる。

## 【 0 0 8 6 】

まず、Berlekamp-Massey法に関しては、アルゴリズムの中にやはり複数の条件分岐を含むことになることが不可避である。したがって、これを組み合わせ回路に展開するときには上述した理由と同じ理由で回路サイズは組み合わせ的に増大する。一方、Euclid法においては多項式の乗除算がアルゴリズムの基本となるが、割り算の分母に現れる多項式の次数が前もってわからないために、ここに条件分岐の入る余地がある。また、この条件分岐に起因してBerlekamp-Massey法の場合と同様の組み合わせ的な回路規模の増大を生じさせてしまうことになる。

## 【 0 0 8 7 】

B. < 組み合わせ回路に適したYule-Walker方程式および誤り位置多項式の計算法の方針 >

上述したように、Levinson(-Durbin)法、Berlekamp-Massey法、Euclid法は共に条件分岐を含むため、組み合わせ回路化という観点からは問題がある。そこで、Yule-Walker方程式の解法を組み合わせ回路で実現するためには場合分けのないアルゴリズムを見出すことが必要であり、これが、本発明のアルゴリズムにおける本質的な方針として与えられる。

## 【 0 0 8 8 】

この場合、上述のアルゴリズムとして、リード・ソロモン符号の復号化で知られているPeterson法を用いることができる。Peterson法のアプローチはYule-Walker方程式を直接解くことになる。この際、Yule-Walker方程式の解は次のようにCramerの公式を用いて行列式の形で表示することができる。

【0089】

$$\Lambda_i^{(l)} = \frac{\tilde{\Lambda}_i^{(l)}}{\tilde{\Lambda}_0^{(l)}}, \quad i = 1, \dots, l$$

$$\tilde{\Lambda}_0^{(l)} = \begin{vmatrix} S_0 & S_1 & \cdots & S_{l-1} \\ S_1 & S_2 & \cdots & S_l \\ \vdots & \vdots & \ddots & \vdots \\ S_{l-1} & S_l & \cdots & S_{2l-2} \end{vmatrix}$$

$$\tilde{\Lambda}_i^{(l)} = \begin{vmatrix} S_0 & S_1 & \cdots & S_{l-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{l-i-1} & S_{l-i} & \cdots & S_{2l-i-2} \\ S_{l-i+1} & S_{l-i+2} & \cdots & S_{2l-i} \\ \vdots & \vdots & \ddots & \vdots \\ S_l & S_{l+1} & \cdots & S_{2l-1} \end{vmatrix}, \quad i = 1, \dots, l-1$$

$$\tilde{\Lambda}_l^{(l)} = \begin{vmatrix} S_1 & S_2 & \cdots & S_l \\ S_2 & S_3 & \cdots & S_{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_l & S_{l+1} & \cdots & S_{2l-1} \end{vmatrix}.$$

したがって、それぞれの

$$l = 1, \dots, t$$

に対して、行列式  $\Lambda^{(1)}_0$  を求め、誤りの個数  $e$  に対して、行列式、

$$\tilde{\Lambda}_i^{(e)}, \quad i = 1, \dots, e$$

を計算すればよいことになる。

【0090】

しかしながら、行列式の展開をそのまま回路として実現すると、 $t$  が増大するにしたがって必要となる乗算器の個数が飛躍的に増大するので同様に直接的な適用は困難である。このため、本発明においては、Hankel行列の帰納的な構造を利

用して計算量の削減を行う。このためのアプローチとしてKatayama-Moriokaによる  $\hat{\Lambda}^{(1)}_i$  の計算と従来の方法と対比して説明する。

【0091】

まず、Katayama-Moriokaの中での  $\hat{\Lambda}^{(1)}_i$  の計算アルゴリズムを  $l=1$  から  $l=4$  まで書き下すと図14に示す形態となる。

【0092】

次にHankel行列の行列式を帰納的なアプローチで計算する別な手法として、Kogaによる方法を対比のために説明する。Kogaによる方法によれば、まず、次の新たな誤り位置多項式、

$${}_{i;i+2u}D(X, Y)$$

が定義されている。ここで、

$${}_{i;i+2u}D(X, Y) = \begin{pmatrix} S_i + YX^i & S_{i+1} + YX^{i+1} & \cdots & S_{i+u} + YX^{i+u} \\ S_{i+1} + YX^{i+1} & S_{i+2} + YX^{i+2} & \cdots & S_{i+u+1} + YX^{i+u+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i+u} + YX^{i+u} & S_{i+u+1} + YX^{i+u+1} & \cdots & S_{i+2u} + YX^{i+2u} \end{pmatrix}.$$

である。

【0093】

そして、この新たな誤り位置多項式を計算するため、 $i$  番目のシンδροーム  $S_i$  を  $(1, 1)$  成分に持つようなHankel行列

$${}_{i;i+2u}Q$$

を考え、

$${}_{i;i+2u}Q$$

から複数の行と列を対称に抜き去った行列式をQ行列式と定義する。Q行列式は、一般に対角成分に現れるシンδροームの添え字番号を左上から順に指定してやれば唯一に定まるものである。ここで、Q行列式を添え字の列、



$$[a_1, a_2, \dots, a_p]$$

で表すことができる。Kogaによる方法では、Q行列式を用いて誤り位置多項式、

$$i; i+2u D(X, Y)$$

を計算するアルゴリズムが提示されている。

【0094】

上述した従来手法は、いずれにしても次に述べるような問題点を抱えている。まず、従来例1の中で用いられた  $\Lambda^{(1)}_i$  の計算アルゴリズムに関しては計算すべき行列式の非対称性に起因して展開の右辺に次々と新しい項が出現し、その結果として行列のサイズが大きくなるにしたがって必要とされる乗算器の個数は組み合わせ的に増大することになる。そこで、このような組み合わせ的発散の程度がなるべく小さいアルゴリズムが望ましい。

【0095】

次にKogaのアルゴリズムについてであるが、Kogaの定義したQ行列式は対称な行列式であり、このことによって乗算器数の削減が実現されている。しかしながら、Kogaのアルゴリズムは、最小距離が偶数の場合のBCH符号もしくはリード・ソロモン符号でしか適用することができないという制限がある。Kogaによれば、この制限を緩和できる場合もあることを開示しているものの、緩和できる例としては、単にbinary narrow sense BCH符号の場合に限定されてしまうことになる。

【0096】

本発明においては、光通信システムへの適用を行うため、(255, 239)リード・ソロモン符号(最小距離=17)の復号化を組み合わせ回路で効率よく実現することが方針として要求されることになる。このため、何らかの手法を用いて最小距離の偶数、奇数にかかわらず効率的な計算を実行させることができるアルゴリズムが必要となる。

【0097】

C. <本発明において使用される用語の定義>

以下に、本発明のアルゴリズムを詳細に説明する前に、本発明において使用する各タームを明確にするために説明を行う。

### (1) シンドローム

一般に、ガロア拡大体  $GF(2^m)$  の原始元を  $a$  とし、 $h < 2^m - 1$  を正整数とすると、

$$G(x) = (x-1)(x-a)(x-a^2)\cdots(x-a^{h-1})$$

を生成多項式とする、符号長が  $n = 2^m - 1$  の  $2^m$  元巡回符号を  $GF(2^m)$  上のリード・ソロモン符号として定義する。すなわち、 $k = n - h$  とし  $k$  個の送信記号を係数にもつ  $k$  次多項式を  $M(x)$  とするとき、 $M(x)$  に  $x^{n-k}$  を乗算し、その結果を次のように  $G(x)$  で除算し、剰余  $R(x)$  を算出する。

【0098】

$$M(x)x^{n-k} = Q(x)G(x) + R(x)$$

ついで、長さ  $n$  の符号化された系列を係数に持つ多項式(送信多項式)を

$$W(x) = M(x)x^{n-k} - R(x) = Q(x)G(x)$$

で定義する。このとき、符号化された送信系列は左端に  $k$  個の情報記号を持ち、それらに  $h = n - k$  個の検査記号が続く組織符号の形になっている。リード・ソロモン符号の最小距離  $d_{\min}$  は、 $d_{\min} = h + 1$  で与えられる。また、訂正可能な誤りの最大個数  $t$  は  $t = \lfloor h/2 \rfloor$  で与えられる。

【0099】

一方、受信された系列よりもとの送信系列を推定する復号化のアルゴリズムは以下のように与えられる。

### (2) シンドロームの計算と誤りの検出

ここで、1 個の誤りが生じたとし、それらの位置を  $i_0, \dots, i_{l-1}$ 、誤りの値を  $E_{i_0}, \dots, E_{i_{l-1}}$  とする。 $E_{i_0}, \dots, E_{i_{l-1}}$  を係数に持つ多項式を、

$$E(x) = E_{i_0}x^{i_0} + \cdots + E_{i_{l-1}}x^{i_{l-1}}$$

で定義すると、受信系列  $b_0, \dots, b_{n-1}$  を係数とする多項式は、

$$Y(x) = \sum_{i=0}^{n-1} b_i x^i = W(x) + E(x)$$

で与えられる。 $Y(x)$  を受信多項式と定義する。

次に、受信多項式  $Y(x)$  からシンドローム、

$$S_i = Y(a^i) \in GF(2^m), \quad i = 0, 1, \dots, 2t-1$$

を計算する。ここで、

$$W(a^i) = 0, \quad i = 0, 1, \dots, 2t-1$$

なので、シンドロームは

$$S_i = E(a^i), \quad i = 0, 1, \dots, 2t-1$$

を満たす。したがって、誤りがなければシンドロームはすべて0となり、シンドロームの値によって誤りの有無が判定できることになる。

【0100】

### (3) 誤りの個数と位置の特定

発生した誤りの個数を仮に1個であると仮定し、発生した位置を、

$$i_0, \dots, i_{l-1}$$

と仮定する。すなわち、

$$b_{i_0}, \dots, b_{i_{l-1}}$$

の値が誤っていると仮定する。誤りの個数1と、下記式の誤り位置、

$$i_0, \dots, i_{l-1}$$

を特定するために、

$$a^{-i_0}, \dots, a^{-i_{l-1}}$$

を根として持つ下記多項式、

$$\Lambda^{(l)}(x) = \prod_{k=0}^{l-1} (1 - a^{i_k} x) = 1 + \Lambda_1^{(l)} x + \dots + \Lambda_{l-1}^{(l)} x^{l-1} + \Lambda_l^{(l)} x^l$$

を定義する。上記式中、

$$a^{-i_0}, \dots, a^{-i_{l-1}}$$

を誤りロケータ、 $\Lambda^{(1)}(x)$  を誤り位置多項式という。

【0101】

さらに、

$$\Lambda_1^{(l)}, \dots, \Lambda_l^{(l)}$$

は、誤り位置多項式を  $x$  に関して展開したときの展開係数であり、

$$a^{i_0}, \dots, a^{i_{l-1}}$$

の基本対称式で与えられる。

【0102】

ここで、下記式

$$\Lambda_1^{(l)}, \dots, \Lambda_l^{(l)}$$

は、次の連立一次方程式、

$$\begin{pmatrix} S_0 & S_1 & \dots & S_{l-1} \\ S_1 & S_2 & \dots & S_l \\ \vdots & & \ddots & \vdots \\ S_{l-1} & S_l & \dots & S_{2l-2} \end{pmatrix} \begin{pmatrix} \Lambda_l^{(l)} \\ \vdots \\ \Lambda_1^{(l)} \end{pmatrix} = \begin{pmatrix} S_l \\ \vdots \\ S_{2l-1} \end{pmatrix}$$

を満たす。これは A. で説明した Yule-Walker 方程式に他ならない。この段階では、 $l$  は未知であるが、実際に生じた誤りの個数が  $1 \leq e \leq t$  であるとき、左辺の Hankel 行列は、 $l = e$  の場合は正則であって、 $t \geq l > e$  の場合は非正則となることが知られている。したがって、 $l = 1, \dots, t$  に対して左辺の Hankel

行列の行列式を計算し、値が非ゼロの最大の整数をもって誤りの個数  $e$  と定めればよい。そして  $1 = e$  の場合にこの方程式を解くことにより、誤り位置多項式を求めることができる。

【0103】

本発明において、誤り位置を特定するためには、誤りロケータ、すなわち、誤り多項式  $\Lambda^{(e)}(x) = 0$  の根を求めればよい。このために下記式

$$a^{-i}, i = 0, 1, \dots, n-1$$

を逐次代入して実際に誤り位置多項式の零点となるか否かを調べるという手法を取ることができる。この手法をChien探索という。誤り位置多項式の零点を、

$$a^{-i_0}, \dots, a^{-i_{e-1}}$$

とするとき、 $i_0, \dots, i_{e-1}$  が実際の誤り位置を与える。

【0104】

#### (4) 誤り値の計算

誤り値の算出は、下記式で示されるVandermonde型の連立線形方程式、

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ a^{i_0} & a^{i_1} & \dots & a^{i_{e-1}} \\ \vdots & \vdots & \ddots & \vdots \\ a^{i_0(e-1)} & a^{i_1(e-1)} & \dots & a^{i_{e-1}(e-1)} \end{pmatrix} \begin{pmatrix} E_{i_0} \\ \vdots \\ \vdots \\ E_{i_{e-1}} \end{pmatrix} = \begin{pmatrix} S_0 \\ \vdots \\ \vdots \\ S_{e-1} \end{pmatrix}.$$

を解くことによって得られる。シンδροームを係数とする多項式  $S(x)$  を、

$$S(x) = S_0 + S_1x + S_{2t-1}x^{2t-1}$$

とおく。さらに、

$$\Omega(x) = \Lambda^{(e)}(x)S(x) \mod x^{2t-1}$$

とおく。 $\Omega(x)$  を誤り評価多項式という。この場合に、Vandermonde型の連立線形方程式の解は、

$$E_{i_k} = \frac{\Omega(a^{-i_k})}{\Lambda'(a^{-i_k})}, \quad i = 0, \dots, e-1$$

によって求めることができる。これをForneyのアルゴリズムという。したがって、誤り位置と誤り値とがわかれば、それらを受信した入力デジタル信号から減算することにより、誤りの訂正されたデジタル信号出力を得ることができる。

【0105】

#### D. 本発明のYule-Walker方程式解法アルゴリズム

我々の課題はGF(2<sup>m</sup>)上で定義された次のYule-Walker方程式の解を組み合わせ回路を用いて計算するための効率的なアルゴリズムを見出すことである。

$$\begin{pmatrix} S_0 & \cdots & S_{l-1} \\ \vdots & \ddots & \vdots \\ S_{l-1} & \cdots & S_{2l-2} \end{pmatrix} \begin{pmatrix} \Lambda_l^{(l)} \\ \vdots \\ \Lambda_1^{(l)} \end{pmatrix} = \begin{pmatrix} S_l \\ \vdots \\ S_{2l-1} \end{pmatrix}.$$

ここで、

$$S_0, S_1, \dots, S_{2l-1}$$

は、与えられたGF(2<sup>m</sup>)の元であり、 $\Lambda_i^{(l)}$  <sup>(1)</sup>が未知の量である。

【0106】

本発明においては、このYule-Walker方程式の解をCramerの公式を用いて、図15のように行列式の形で表示し、その帰納的な構造を利用して行列式の効率のよい計算法を求めるものである。

【0107】

図15に示した行列式を計算するために、本発明においては、次のJacobiの公式に注目する。

<Jacobiの公式>

$A = (a_{ij})$  を単位元1をもつ可換環上のn次正方行列とする。Aの(i, j)余因子を $\Delta_{ij}$ とする。添え字の集合、

$$\mu = \{i_1, \dots, i_r, (i_1 < \dots < i_r)\}, \nu = \{j_1, \dots, j_r, (j_1 < \dots < j_r)\}$$

に対する小行列式 $A_{\mu\nu}^{(r)}$ の余因子を $\Delta_{\mu\nu}^{(r)}$ とすると、下記式が成

り立つ。

【0108】

$$\begin{vmatrix} \Delta_{ij_1} & \cdots & \Delta_{ij_r} \\ \vdots & \ddots & \vdots \\ \Delta_{irj_1} & \cdots & \Delta_{irj_r} \end{vmatrix} = (\det A)^{r-1} \Delta_{\mu\nu}^{(n-r)}.$$

本発明では特に、

$$\mu = \{i_1, i_2\}, \nu = \{j_1, j_2\}$$

とした場合に成り立つ下記式、

$$\Delta_{ij_1} \cdot \Delta_{i_2j_2} - \Delta_{i_2j_1} \cdot \Delta_{ij_2} = (\det A) \Delta_{\mu\nu}^{(n-2)}$$

を用いることができる。

【0109】

このJacobiの公式を用いて  $\Lambda^{\text{hat}}_i (1)$  を計算するために次のように考える。

まず、 $\Lambda^{\text{hat}}_i (1+1)$  は下記式の形をしている。

【0110】

$$\bar{N}_0^{(l+1)} = \begin{vmatrix} S_0 & S_1 & \cdots & S_{l-1} & S_l \\ S_1 & \ddots & & \vdots & S_{l+1} \\ \vdots & & \ddots & \vdots & \vdots \\ S_{l-1} & \cdots & \cdots & S_{2l-2} & S_{2l-1} \\ S_l & S_{l+1} & \cdots & S_{2l-1} & S_{2l} \end{vmatrix}.$$

この行列式をよく眺めてみると、 $\Lambda^{\text{hat}}_i (1)$  は、 $\Lambda^{\text{hat}}_i (1+1)$  から  $(1+1-i)$  行および第1列を取り除いたものであり、 $\Lambda^{\text{hat}}_0 (1)$  は  $\Lambda^{\text{hat}}_0 (1+1)$  から第1行および第1列を取り除いたものがわかる。つまり、 $\Lambda^{\text{hat}}_0 (1)$  は、 $\Lambda^{\text{hat}}_i (1)$  が、それぞれ  $\Lambda_i (1+1)$  の  $(1+1, 1+1)$ 、 $(1+1, 1+1-i)$  余因子であるため、Jacobiの公式において、

$$i_1 = j_1 = l+1-i, i_2 = j_2 = l+1.$$

として、

$$\Delta_{l+1,l+1} = \bar{\Lambda}_0^{(l)}, \quad \Delta_{l+1,l+1-i} = \Delta_{l+1-i,l+1} = \bar{\Lambda}_i^{(l)}$$

とおく。さらに、 $\Lambda^{\text{hat}}_0^{(1+1)}$  の  $(1+1-i, 1+1-i)$  余因子を  $\Gamma_i^{(1+1)}$  と定義する。ここで、 $\Gamma_i^{(1+1)}$  の構成を図 16 に示す。

そして、Jacobi の公式を用いることにより、

$$\Gamma_i^{(l+1)} \bar{\Lambda}_0^{(l)} + (\bar{\Lambda}_i^{(l)})^2 = \bar{\Lambda}_0^{(l+1)} \Gamma_{i-1}^{(l)}, \quad i = 1, \dots, l$$

を得る。

【0111】

この公式を用いると、 $\Lambda^{\text{hat}}_i^{(1)}$  の計算は対称行列の行列式である  $\Gamma_i^{(1+1)}$  の計算に帰着される。ただし、 $\Lambda^{\text{hat}}_i^{(1)}$  を求めるには、 $\Gamma_i^{(1+1)}$  の計算に加えて、 $2 \times 1$  個の乗算と 1 個の平方根をとる操作が必要となる。平方根を取る計算と 2 乗を行う計算とは、線形演算として実現できるため加算とほぼ同等のコストで回路として実現可能である。したがって、これらは非線形演算回路である乗算器に比べて非常に小さなコストしか要しない。そこで我々は乗算器にのみ注目し、その個数を問題とする。提案するアルゴリズムの場合、 $GF(2^m)$  は標数が 2 であること、および  $\Gamma_i^{(1)}$  はすべて対称あることから、行列式の余因子展開において対角線に関して非対称な配置より生じる項は必ずキャンセルする。例えば、 $3 \times 3$  の対称行列を例にとって余因子展開を計算してみると、

$$\begin{vmatrix} a & b & c \\ b & d & e \\ c & e & f \end{vmatrix} = adf + ae^2 + b^2f + bec + c^2d + bec = adf + ae^2 + b^2f + c^2d$$

となって、対角線に関して非対称な配置より生じる項  $b e c$  は対称性により必ず 2 度出てくるために、キャンセルされる。このため、本発明のアルゴリズムを乗算器を含む組み合わせ回路に使用した場合には、要求される乗算器の個数が低減できることになる。

【0112】



ここで、

$$\Gamma_i^{(l)}, l = 1, 2, \dots, t+1, i = 0, 1, \dots, t$$

を帰納的に計算するアルゴリズムの一般形は以下のように与えられる。

$$0. \quad \Gamma_1^{(1)} = 1, \quad \Gamma_0^{(2)} = S_0, \quad \Gamma_1^{(2)} = S_1$$

$$1. \quad l > 2, \quad i = 1 \text{ のとき}$$

$$\Gamma_0^{(l)} = S_{2l-4} \Gamma_0^{(l-1)} + \sum_{k=1}^{l-2} S_{2l-4-k}^2 \Gamma_{k-1}^{(l-2)}.$$

$$2. \quad l > 2, \quad i = 1, \dots, l-1 \text{ のとき}$$

まず、アルゴリズムを記述する際の補助的な量を一つ定義する。

【0113】

$$\{i_1, \dots, i_n\}$$

を、添え字の集合とすると、 $\det[\{i_1, \dots, i_n\}]$ を

$$\det[\{i_1, \dots, i_n\}] = \begin{vmatrix} S_{i_1} & S_{i_2} & \cdots & S_{i_n} \\ S_{i_2} & S_{2i_2-i_1} & \cdots & S_{i_2+i_n-i_1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i_n} & S_{i_2+i_n-i_1} & \cdots & S_{2i_n-i_1} \end{vmatrix}$$

と定義する。正確にいうと、 $\det[\{i_1, \dots, i_n\}]$ は、第1行目が、

$$S_{i_1}, \dots, S_{i_n}$$

であって、 $(p, q)$ 成分が下記式、

$$S_{i_p+i_q-i_1}$$

である対称行列の行列式であり、Hankel行列式 $\Lambda_0^{(1)}$ からいくつかの行と列

とを対称に抜き去って得られる行列式である。ここで、 $\det[\{i_1, \dots,$

$i_n\}]\text{を用いると}\Gamma_i^{(1)}\text{は次のように計算される。}$

【0114】

$$\Gamma_i^{(l)} = S_{2l-2} \Gamma_{i-1}^{(l-1)} + \sum_{k=1, k \neq i}^{l-1} S_{2l-2-k}^2 \det[\{0, 1, \dots, l-2\} - \{l-1-i, l-1-k\}]..$$

上記式中、 $\det[\{0, 1, \dots, l-2\} - \{l-1-i, l-1-k\}]$ は、 $\Gamma_i^{(l-1)}$  から対称に2つの  $l-1-i, l-1-k$  行と、2つの  $l-1-i, l-1-k$  列とを抜き去って得られる対称行列の行列式であって、これは  $k=1$  の場合と  $i=1$  との場合には、それぞれ、下記式、

$$\Gamma_{i-2}^{(l-2)}, \Gamma_{k-2}^{(l-2)}$$

に一致することに注意されたい。

【0115】

3. 一般に  $\det[\{i_1, \dots, i_n\}]$  は、次のように計算される。

【0116】

$$\det[\{i_1, \dots, i_n\}] = S_{2i_n-i_1} \det[\{i_1, \dots, i_{n-1}\}] + \sum_{k=1}^{n-1} S_{i_{n-1}+i_k}^2 \det[\{i_1, \dots, i_{n-1}\} - \{i_k\}].$$

E. <本発明のアルゴリズムのリード・ソロモン符号の復号化への適用>

以下に、D. で述べた Yule-Walker 方程式の本発明の解法アルゴリズムを、リード・ソロモン符号へと応用した場合についての実施の形態を説明する。Yule-Walker 方程式自身は、通常は、次元(未知数の個数)は一定のものとして与えられるのであるが、リード・ソロモン符号の復号化の場合、次元(誤りの個数に対応している)も未知なので、これも含めて決定しなければならないことになる。

【0117】

(1)  $\Gamma_i^{(1)}$  の計算

シンδροームの系列、

$$S_0, S_1, \dots, S_{2t-1}$$

が与えられたとき、D. において説明したアルゴリズムに従い、

$$\Gamma_i^{(l)}, l = 1, 2, \dots, t+1, i = 0, \dots, t$$

を計算する。この計算の過程で、

$$\tilde{\Lambda}_0^{(l)} = \Gamma_0^{(l+1)}, l = 1, \dots, t$$

も、同時に計算される。なお、本発明者らは、リード・ソロモン符号の高速復号化を行うために、組み合わせ回路としての実現を念頭に置いているが、本発明においては組み合わせ回路と共に使用することに限定されるものではなく、本発明の誤り訂正アルゴリズムは、順序回路を用いて誤り訂正装置として実装することも可能である。

【0118】

## (2) 誤りの個数の決定

実際に生じた誤りの個数を、 $e$  で表す。ここで、 $e$  は

$$\tilde{\Lambda}_0^{(l)} = \Gamma_0^{(l+1)}, l = 1, \dots, t$$

の値から、 $\Lambda^{\text{hat}}_0^{(1)} \neq 0$  を満たす最大の  $l$  として求めることができる。

【0119】

## (3) 誤り位置多項式の決定

誤りの個数を決定した結果、もし  $e < t$  が判明した場合には、 $\Lambda^{\text{hat}}_0^{(e+1)} = 0$  なので、本発明のアルゴリズムに従い、

$$\tilde{\Lambda}_i^{(e)} = \sqrt{\Gamma_i^{(e+1)} \tilde{\Lambda}_0^{(e)}}, i = 1, \dots, e.$$

のように簡単化することができる。誤りロケータは、誤り位置多項式の零点であるので、誤り位置多項式の係数の定数倍によって不変である。したがって、 $\Lambda^{\text{hat}}_i^{(e)}$  の代わりに、下記式

$$\sqrt{\Gamma_i^{(e+1)}}$$

を用いることができる。つまり、上式に現れている掛け算は必要ない。一方、 $e = t$  であることが判明した場合には、

$$\tilde{\Lambda}_i^{(e)} = \sqrt{\Gamma_i^{(e+1)} \tilde{\Lambda}_0^{(e)} + \tilde{\Lambda}_0^{(e+1)} \Gamma_{i-1}^{(e)}}, i = 1, \dots, e.$$

に従って誤り位置多項式を計算することになる。このとき、我々のアルゴリズムでは、最小距離が奇数 ( $= 2t + 1$ ) の場合には、計算できない  $S_{2t}$  が見かけ上必要になるように見える。しかし、本発明の式は、シンδροームに関する恒等式であるため、 $S_{2t}$  に関する恒等式にもなっている。そして、計算すべき  $\Lambda^{\text{hat}}_i(t)$  は、シンδροーム  $S_{2t}$  を含まないので、 $\Lambda^{\text{hat}}_0(t+1)$  と、 $\Gamma_i(t+1)$  の余因子展開に現れる  $S_{2t}$  は必ずキャンセルする。具体的にいうと、 $\Gamma_i(t+1)$  を余因子展開したときに現れる項のうち、 $S_{2t}$  を含むものは、 $\Gamma_{i-1}(t) S_{2t}$  であるから、 $\Gamma_i(t+1) \Lambda^{\text{hat}}_0(t)$  を展開したとき、 $S_{2t}$  を含む項は  $\Lambda^{\text{hat}}_0(t) \Gamma_{i-1}(t) S_{2t}$  である。一方、 $\Lambda^{\text{hat}}_0(t+1)$  を余因子展開したときに現れる項のうち、 $S_{2t}$  を含むものは、 $\Lambda^{\text{hat}}_0(t) S_{2t}$  であるから、 $\Lambda_0(t+1) \Gamma_{i-1}(t)$  を展開したとき  $S_{2t}$  を含む項は、 $\Lambda^{\text{hat}}_0(t) \Gamma_{i-1}(t) S_{2t}$  である。したがって、両者は必ずキャンセルする。

【0120】

こうして、 $\Lambda_0(t+1)$  と、 $\Gamma_i(t+1)$  を余因子展開したときに現れる項のうち、 $S_{2t}$  を係数にもつ項は計算する必要がないことが示される。このようにして、本発明のアルゴリズムは、任意の最小距離をもつリード・ソロモン符号に適用することができることとなる。同時に、乗算器数の削減という観点からも、 $S_{2t}$  を含む項の乗算が必要ないので、Kogaアルゴリズムと比較しても本発明のアルゴリズムは、優位になる。また、上述したように、平方根を算出する計算は、加算とほぼ同等のコストで回路として実現可能であり、これは乗算器に比べて非常に小さなコストしか要しないものである。

【0121】

F. <リード・ソロモン符号の復号化への適用例>

上述したEにおいて説明した本発明の誤り訂正アルゴリズムを  $t = 4$  のリード・ソロモン符号の復号化へ適用する実施の形態について、以下に説明する。 $t = 4$  の場合には、本発明により、以下の各式が決定される。ただし、簡単のため、

$$\det[\{i_1, \dots, i_n\}] = \det i_1 \cdots i_n$$

として表す。

(1)  $\Gamma_i^{(1)}$ ,  $i = 0, \dots, l-1, \dots, 5$  の計算  
本発明による計算結果を、図 17 に示す。

【0122】

(2) 誤りの個数の決定

81) で算出された  $\Gamma_i^{(1)}$  によって、

$$\tilde{\Lambda}_0^{(l)} = \Gamma_0^{(l+1)}, l = 1, \dots, 4$$

が求まるので、

$$\tilde{\Lambda}_0^{(l)} \neq 0$$

を満たす最大の  $l$ ,  $l = 1, 2, 3, 4$  として誤りの個数  $e$  が決定できる。

【0123】

(3) 誤り位置多項式の決定

(2) による計算により、例えば  $e = 2$  であることが判明した場合には、誤りロケータ、

$$a^{i_0}, a^{i_1}$$

は、次の代数方程式、

$$\sqrt{\tilde{\Lambda}_0^{(2)}} + \sqrt{\Gamma_1^{(3)}} x + \sqrt{\Gamma_2^{(3)}} x^2 = 0$$

を解くことによって求められる。一方、 $e = 4$  であることが判明した場合には、上述したように、

$$\tilde{\Lambda}_i^{(4)} = \sqrt{\Gamma_i^{(5)} \tilde{\Lambda}_0^{(4)} + \tilde{\Lambda}_0^{(5)} \Gamma_{i-1}^{(4)}}, \quad i = 1, 2, 3, 4$$

によって求めることができる。ただし、

$$\Gamma_i^{(5)}, \Gamma_0^{(6)} = \tilde{\Lambda}_0^{(5)}$$

の計算において、シンδροーム  $S_8$  を含む項の計算は上述したように不要である

【0124】

図18は、上述した本発明の誤り訂正アルゴリズムの概略的なフローチャートを示した図である。本発明の誤り訂正アルゴリズムにおいては、まず、ステップ200において、シンδροーム  $S_0, \dots, S_{2t-1}$  が入力され、ステップ201において、誤り多項式  $\Gamma$  が算出される。 $\Gamma_0^{(2)}, \dots, \Gamma_0^{(t+1)}$  が求められた段階で、ステップ202において、 $\Lambda^{\text{hat}}_0^{(m)} = \Gamma_0^{(m+1)} \neq 0$  を満たす最大の整数  $m$ 、として誤りの個数を決定する。ステップ203においては、誤りの個数  $e$  が、最大の誤りの数に等しいか否かが判断され、 $e=t$  の場合 (yes) には、 $\Gamma_0^{(e+1)} = \Lambda^{\text{hat}}_0^{(e)}, \dots, \Gamma_e^{(e+1)}, \Gamma_0^{(e+2)} = \Lambda^{\text{hat}}_0^{(e+1)}$  として、ステップ204において、誤り値を算出する。また、 $e \neq t$  の場合 (no) には、 $\Gamma_0^{(e+1)} = \Lambda^{\text{hat}}_0^{(e)}, \dots, \Gamma_e^{(e+1)}$  のみを使用して、ステップ205において誤り値を計算し、ステップ206において、 $\Lambda^{\text{hat}}_0^{(e)}, \dots, \Lambda^{\text{hat}}_e^{(e)}$  を得る。

【0125】

G. 本発明のアルゴリズムを誤り位置多項式の計算へ適用した場合の計算回路  
図19に本発明で提案するアルゴリズムに基づいた誤り位置多項式の計算回路のブロック図を示す。図20に示される本発明のアルゴリズムを使用した誤り位置多項式の計算回路は、概ね  $\{\Gamma_i^{(m)}\}$  計算ブロック100と、誤りの個数を計算する回路ブロック102と、誤り位置多項式の決定を行う回路ブロック104とを含んで構成されている。

【0126】

図19の各ブロックの機能を説明すると、回路ブロック100には、例えば順序回路を使用して入力デジタル信号から算出されたシンδροームのシリーズが入力される。回路ブロック100においては、これらのシンδροームから、

$$\Gamma_i^{(m)}, m = 1, 2, \dots, t+1, i = 0, \dots, t$$

が、本発明のアルゴリズムにしたがって帰納的に計算される。これはアルゴリズム

μの詳細の(1)に対応する。

【0 1 2 7】

次に、回路ブロック 1 0 2 においては、計算された、

$$\Gamma_0^{(m)}, m = 1, 2, \dots, t+1$$

の値から、誤りの個数  $e$  を算出し、 $e$  の値にそれぞれ対応する、

$$\Gamma_i^{(e+1)}, i = 0, \dots, e$$

を出力する。 $e = t$  の場合には、これらに加えて、さらに、

$$\Gamma_0^{(t+2)} = \tilde{\Lambda}_0^{(t+1)}$$

も出力される。これはアルゴリズムの詳細の(2)に対応する。回路ブロック 1 0 4 では、

$$\Gamma_i^{(e+1)}, i = 0, \dots, e$$

の値を用いて、誤り位置多項式の係数の計算を実行する。これはアルゴリズムの詳細の(3)に対応するプロセスに従って実行される。

【0 1 2 8】

なお、本発明においては、リード・ソロモン符号の高速復号化を行うために、組み合わせ回路としての実現を念頭に置いているが、提案するアルゴリズムを回路サイズの縮小を目的として順序回路を用いて実現することも可能である。

【0 1 2 9】

H. <リード・ソロモン符号の復号化に適用した場合の回路サイズ>

本発明のアルゴリズムを、リード・ソロモン符号の復号化へ適用した場合の、回路サイズについて以下に説明する。上述したとおり、平方根を算出する計算と2乗を行う計算とは、加算とほぼ同等のコストで回路として実現可能であり、これらは乗算器に比べて非常に小さなコストしか要しない。そこで我々は乗算器にのみ注目し、その個数を検討する。

【0 1 3 0】

表4は、本発明のアルゴリズムによって必要とされる乗算器の個数を  $t = 1$  から  $t = 8$  までの範囲で示したものである。この図には比較のため、従来例1および従来例2の計算アルゴリズムによる乗算器数も合わせて記載した。

【0131】

【表4】

乗算器数の比較

訂正可能な誤りの個数 $t$	1	2	3	4	5	6	7	8
従来例1のアルゴリズム	0	3	17	48	117*	255*	548*	1111*
従来例2のアルゴリズム	2	9	22	49	98	189	351	640
本発明のアルゴリズム	2	7	21	46	94	179	331	597

注 \* 印のついた数値は推定値

【0132】

表4に示されるように、今回提案するアルゴリズムは、乗算器の個数の観点からみてKogaにより提案されたアルゴリズム（従来例2）よりもすべての  $t$  で優れていることが示されている。また、光通信分野への応用においては、特に(255、239)リード・ソロモン符号（ $t = 8$ ）の復号化が重要であるが、これは最小距離が奇数（ $= 17$ ）であるためにKogaアルゴリズムは適用できない。しかしながら、本発明のアルゴリズムは、任意の最小距離のリード・ソロモン符号に適用可能であるため、(255、239)リード・ソロモン符号にも用いることができる。これを、表5に示す。

【0133】



【表 5】

Koga アルゴリズムと本発明のアルゴリズムの適用範囲の比較

符 号 の 最 小 距 離	...	15	16	17	18	...
Koga アル ゴリズム	...	×	○	×	○	...
提案するアル ゴリズム	...	○	○	○	○	...

(×はアルゴリズムが適用できないことを表し、○はできることを意味する。Koga アルゴリズムが偶数の最小距離の場合しか適用できないのに対し、提案するアルゴリズムは任意の最小距離で適用可能。ITU で標準化された (255, 239) リード・ソロモン符号は最小距離が 17 の場合である。)

## 【0134】

一方、従来例 1 (Katayama-Morioka) における計算アルゴリズムも任意の最小距離のリード・ソロモン符号に適用可能であるが、乗算器の個数の観点から比較すると今回提案するアルゴリズムは、 $t$  が 4 以上では、従来例 1 のアルゴリズムよりもより少ない数の乗算器しか必要としない。特に、 $t = 8$  の場合には、本発明のアルゴリズムは、約 40% の乗算器の削減を実現することが可能となること示された。具体的な回路サイズで比較すると、 $t = 8$  のとき、誤り値の計算に要するゲート数は、現在約 10 K ゲートであるのに対して、従来例 1 では、約 80 K ゲートを要するものと考えられる。しかしながら、本発明におけるアルゴリズムを用いると、誤り多項式の計算を約 40 K ゲートまで削減することが可能となる。

## 【0135】

図 20 には、本発明の誤り訂正装置の概略ブロック図を示す。図 20 に示された誤り訂正装置は、入力デジタル信号を受信して符号化する符号化ブロック 110 と、符号化された入力デジタル信号  $ID$  が入力され、シンδροームを算出するための入力ブロック 112 と、復号回路を含む処理ブロック 114 と、誤り位置および誤り値との出力を使用して誤りが訂正された出力デジタル信号  $OD$  を出力する出力ブロック 116 とから構成されている。符号化ブロック 110 に

は、インターリーブされた波長多重通信により送信された入力デジタル信号が入力され、例えば、リード・ソロモン符号化され、入力ブロックへと符号化されたデジタル信号を入力している。入力ブロック 1 1 2 は入力デジタル信号から順序回路を使用してシンドロームを算出し、その出力を処理ブロック 1 1 4 へと送っている。

## 【 0 1 3 6 】

処理ブロック 1 1 4 には、本発明のアルゴリズムを実行する復号機能が含まれていて、誤り位置と誤り値とを算出する。算出された誤り位置と、誤り値は、出力ブロック 1 1 6 へと出力され、誤りが訂正され、出力デジタル信号として出力を行っている。上述した誤り訂正回路は、複数のハードウェアからなる誤り訂正装置として構成することができる他、半導体技術を利用し、各機能ブロックをシリコン・ウエハ上に構成した A S I C といった半導体デバイスとして構成することができることはいうまでもないことである。さらには、本発明のアルゴリズムは、誤り訂正装置のファームウェアとして実装することもできるし、またフロッピーディスク、ハードディスク、光ディスク、光磁気ディスクといった記憶媒体に記録されたコンピュータ可読なプログラムとすることもできる。また、本発明のプログラムは、例えばオブジェクト指向のいかなる言語や、例えば C 言語といったプログラミング言語により、記述し、上述した記録媒体内に保持させて使用することができる。

## 【 0 1 3 7 】

上述したように、本発明によれば、高速の光通信分野において特に効果的に誤りを訂正することを可能とする復号回路、該誤り訂正方法を提供することができる。

## 【図面の簡単な説明】

## 【図 1】

従来の復号回路を示した図。

## 【図 2】

従来の光通信用誤り訂正回路を示した図。

## 【図 3】

従来の回路規模と、データ転送速度とをプロットした図。

【図 4】

さらに別の従来の復号回路を示した図。

【図 5】

さらに別の従来の復号回路を示した図。

【図 6】

本発明の復号回路の実施の形態の概略図。

【図 7】

従来の構成の乗算回路を示した図。

【図 8】

従来の構成の乗算回路を示した図。

【図 9】

従来の構成の乗算回路を示した図。

【図 1 0】

図 7 に示す乗算回路に本発明を適用した実施の形態を示した図。

【図 1 1】

図 8 に示す乗算回路に本発明を適用した実施の形態を示した図。

【図 1 2】

図 9 に示す乗算回路に本発明を適用した実施の形態を示した図。

【図 1 3】

本発明の乗算回路を使用した場合の回路サイズと乗算回路数とをプロットした図。

【図 1 4】

従来の誤り多項式を示した図。

【図 1 5】

本発明における Yule-Walker 方程式の定式化を示した図。

【図 1 6】

本発明における  $\Gamma_i^{(i+1)}$  の詳細な構成を示した図。

【図 1 7】

本発明におけるリード・ソロモン符号の復号の詳細な計算結果を示した図。

【図 1 8】

本発明の誤り訂正アルゴリズムの概略フローチャート。

【図 1 9】

本発明のリード・ソロモン符号の復号回路の概略構成を示した図。

【図 2 0】

本発明の誤り訂正装置の構成を示す概略ブロック図。

【符号の説明】

- 1 0 … 入力部
- 1 2 … 処理部
- 1 4 … 出力部
- 1 6 … シンドローム算出部
- 1 8 … 位置多項式係数算出部
- 2 0 … 誤り値多項式係数算出部
- 2 2 … レジスタ
- 2 4 … ANDゲート
- 2 6 … XORゲート
- 2 8 a, 2 8 b … i m n バッファ
- 4 0 … 乗算回路
- 4 2 … 乗算回路
- 4 5 a, 4 5 b … 出力
- 4 6 … 乗算器群
- 4 7 … 加算器群
- 5 2 … XOR群
- 5 4 … XOR群
- 5 6 … XOR群
- 6 0 … 下流側 XOR群
- 6 0 a, 6 0 b … XOR群
- 6 2 … 出力

64…出力

66…入力

68、70、72、77、78…入力

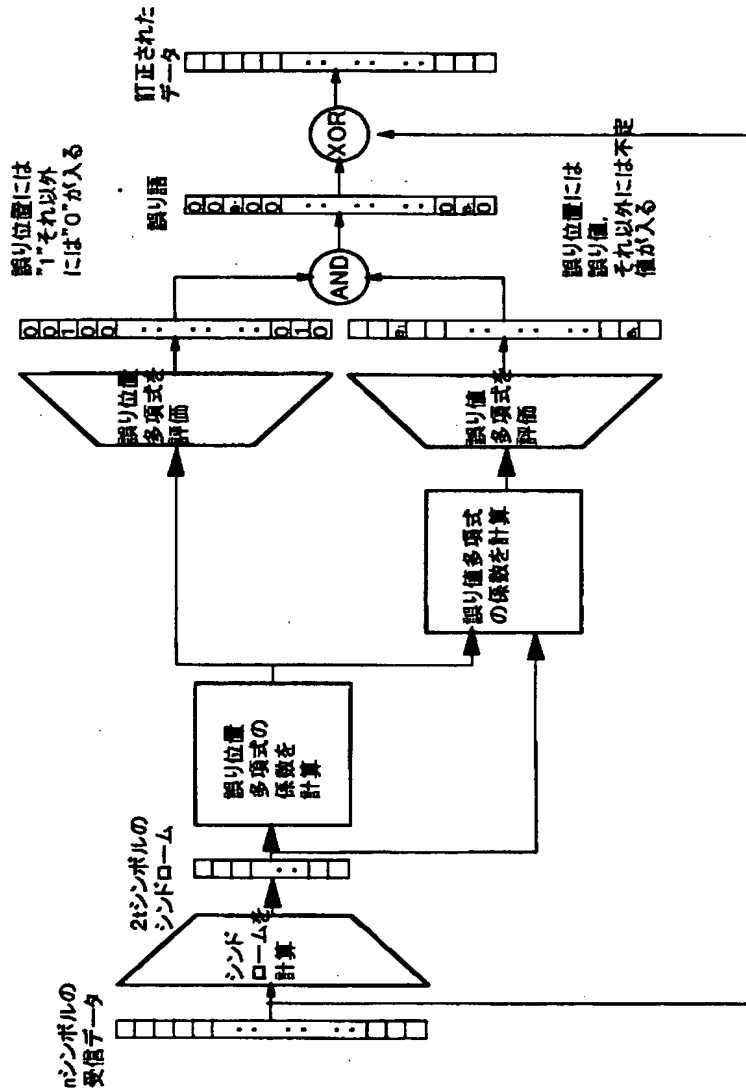
80…AND群

82…AND部

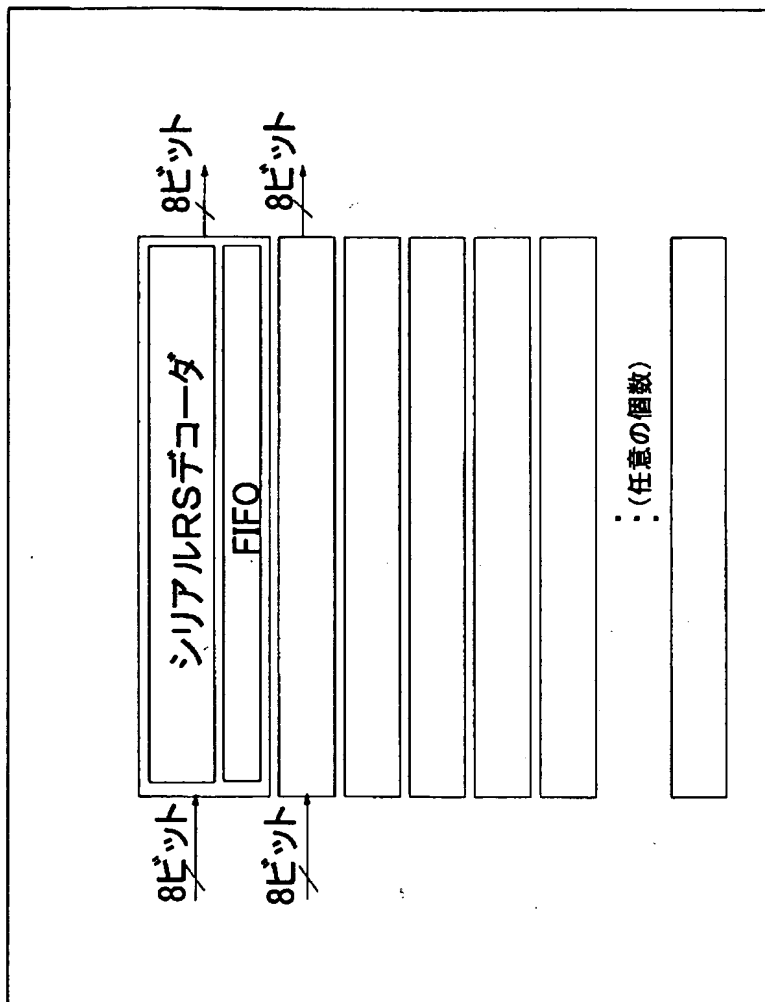
84…加算器

【書類名】 図面

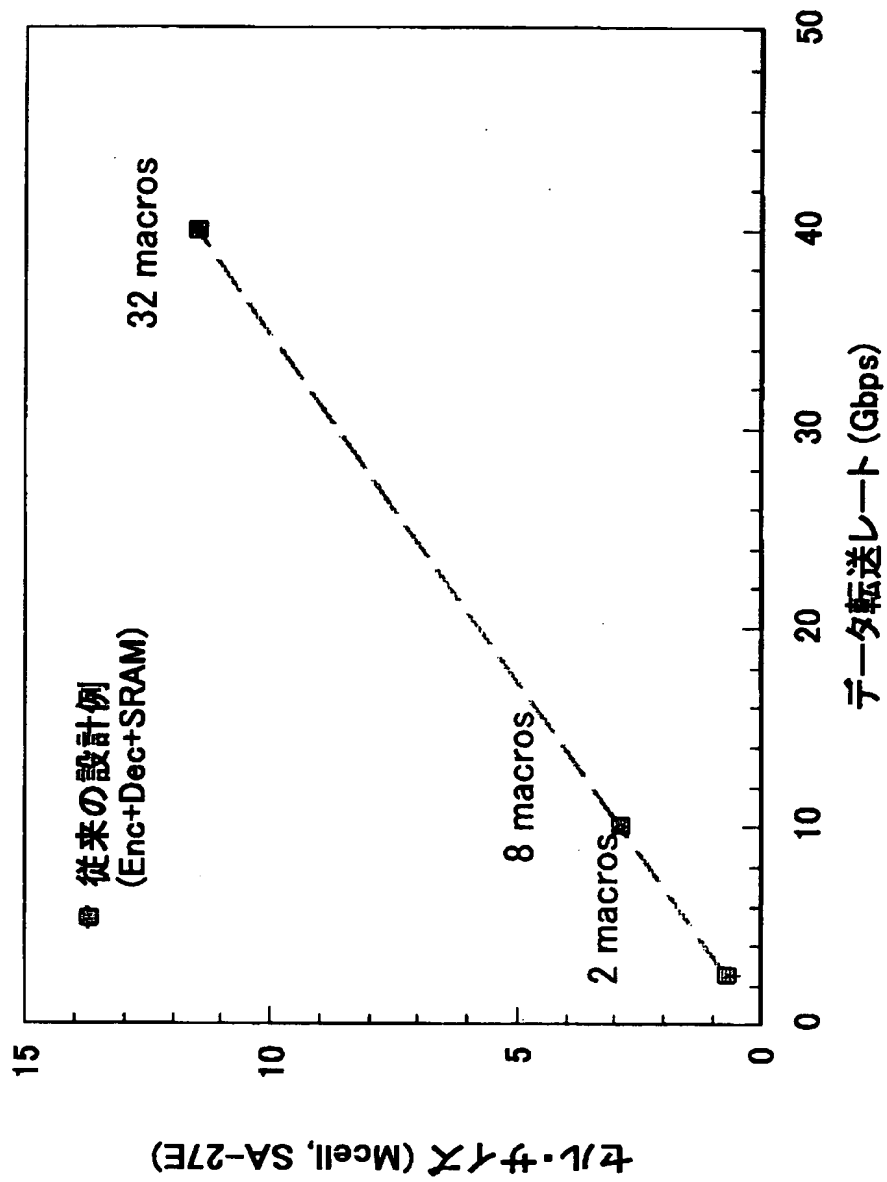
【図 1】



【図 2】

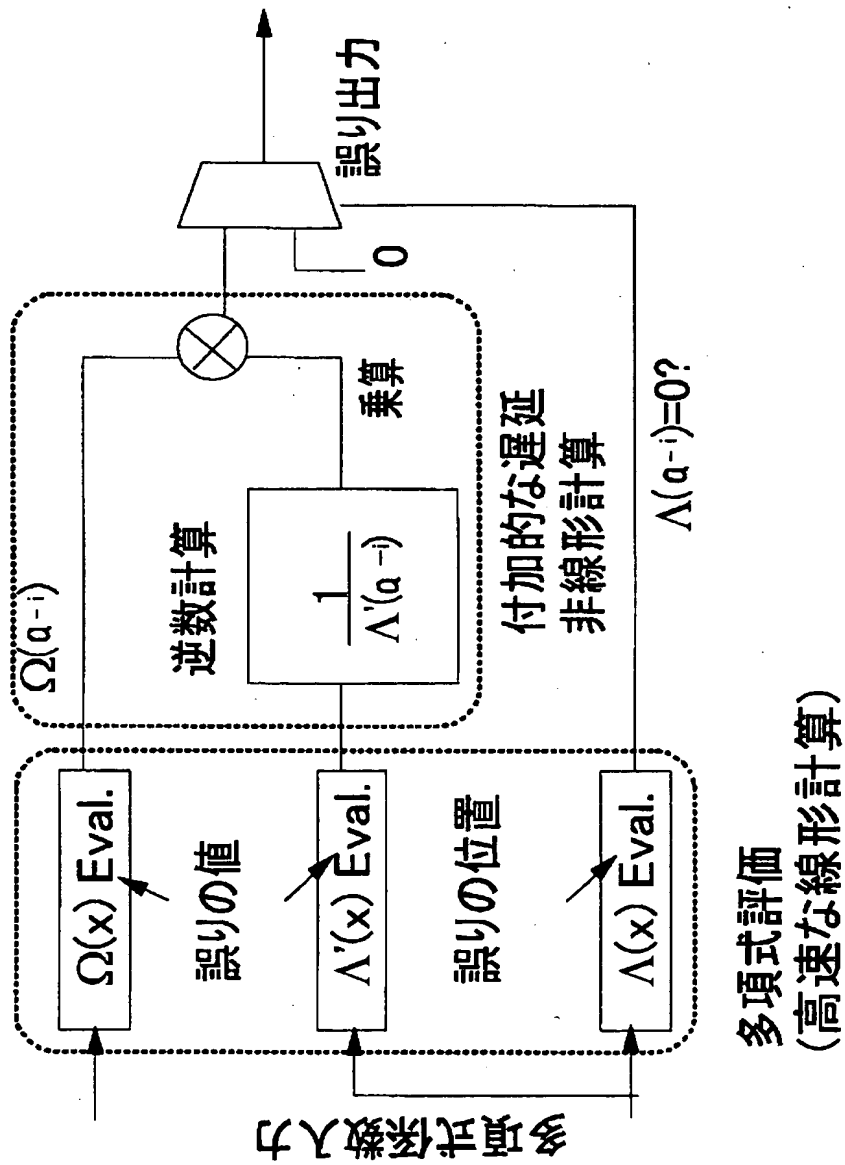


【図3】

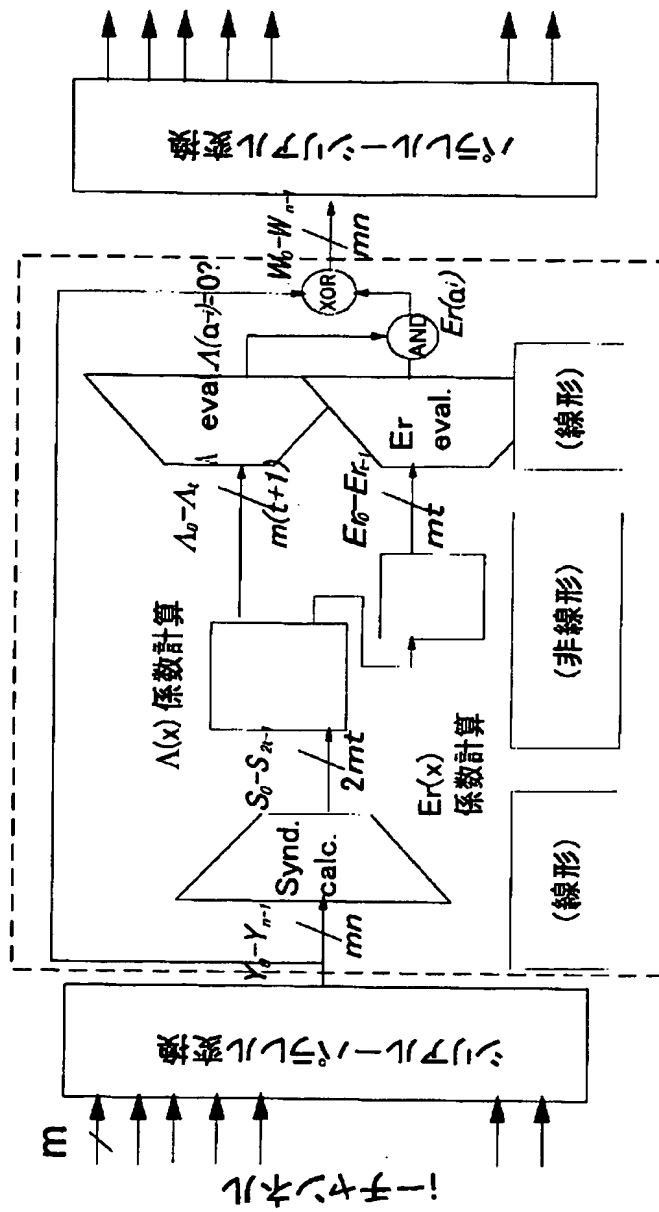




【図4】

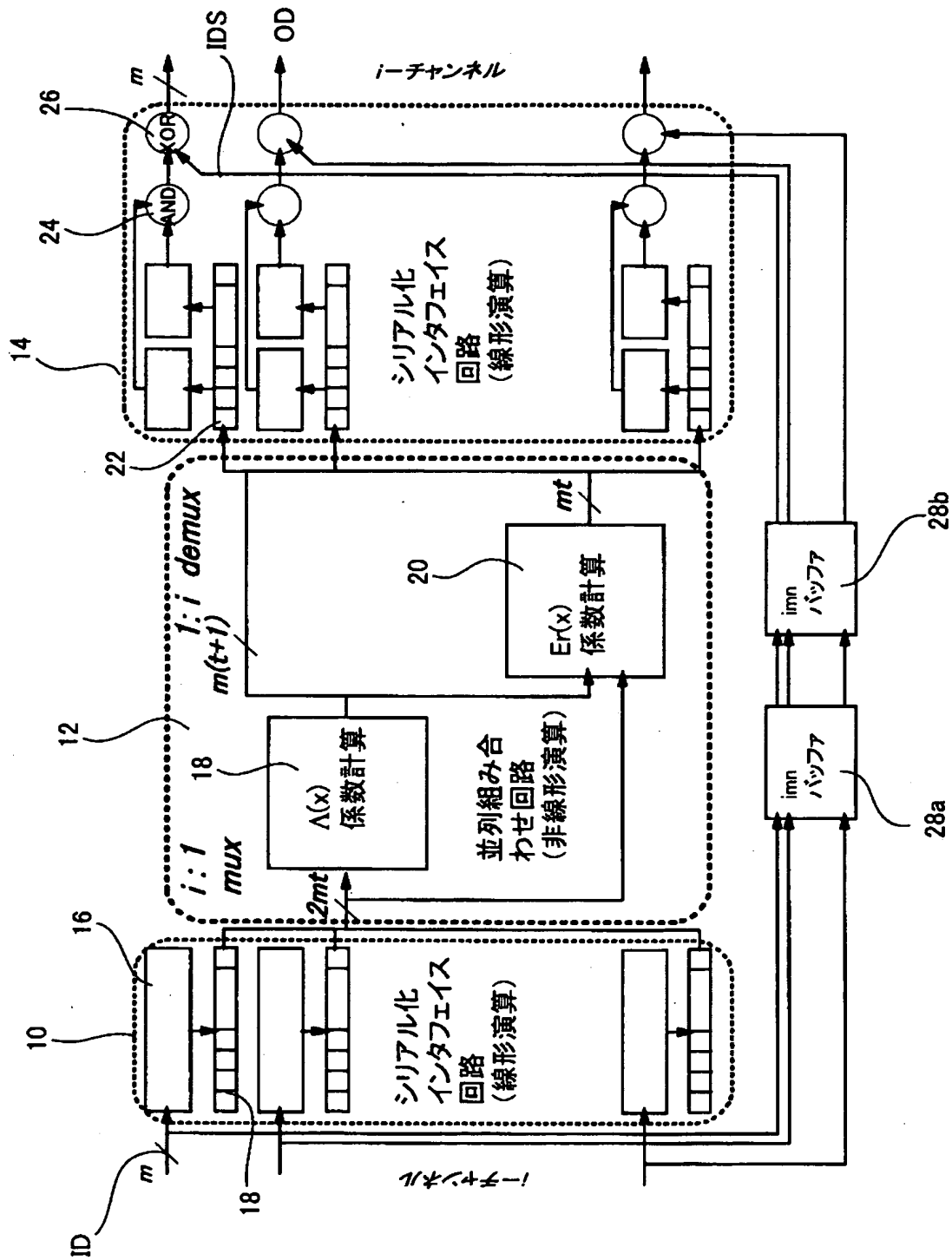


【図 5】

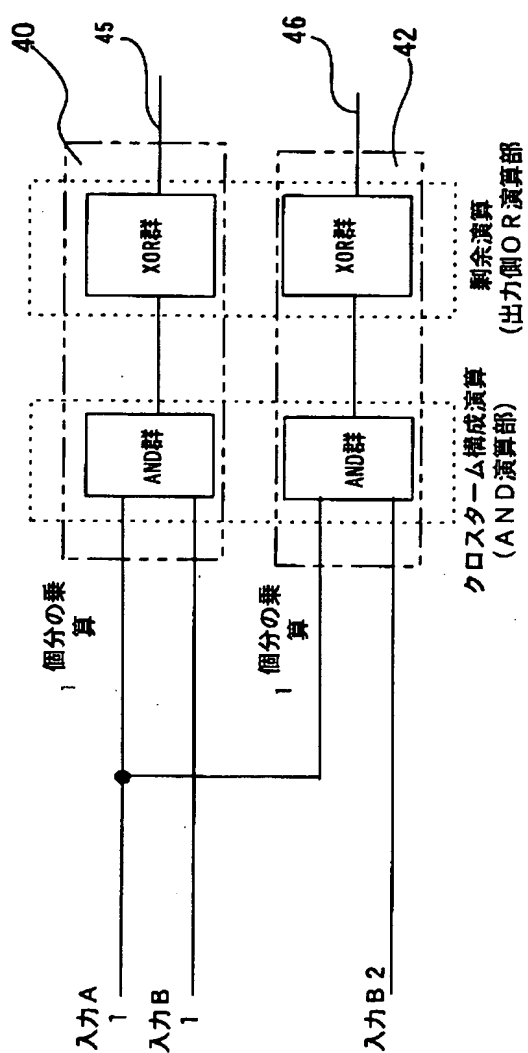


組み合わせ回路

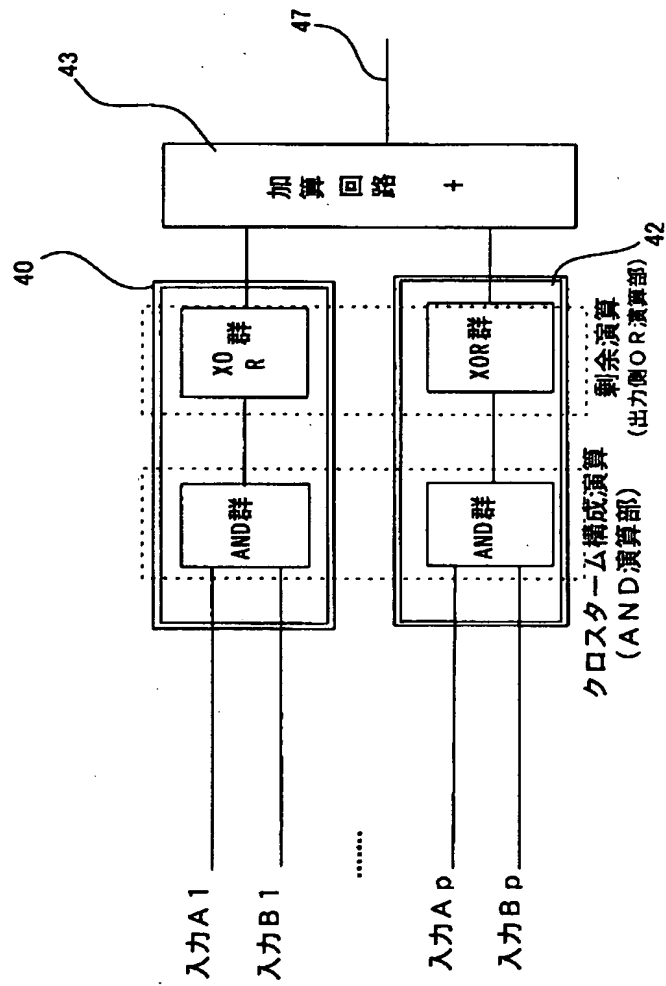
【図6】



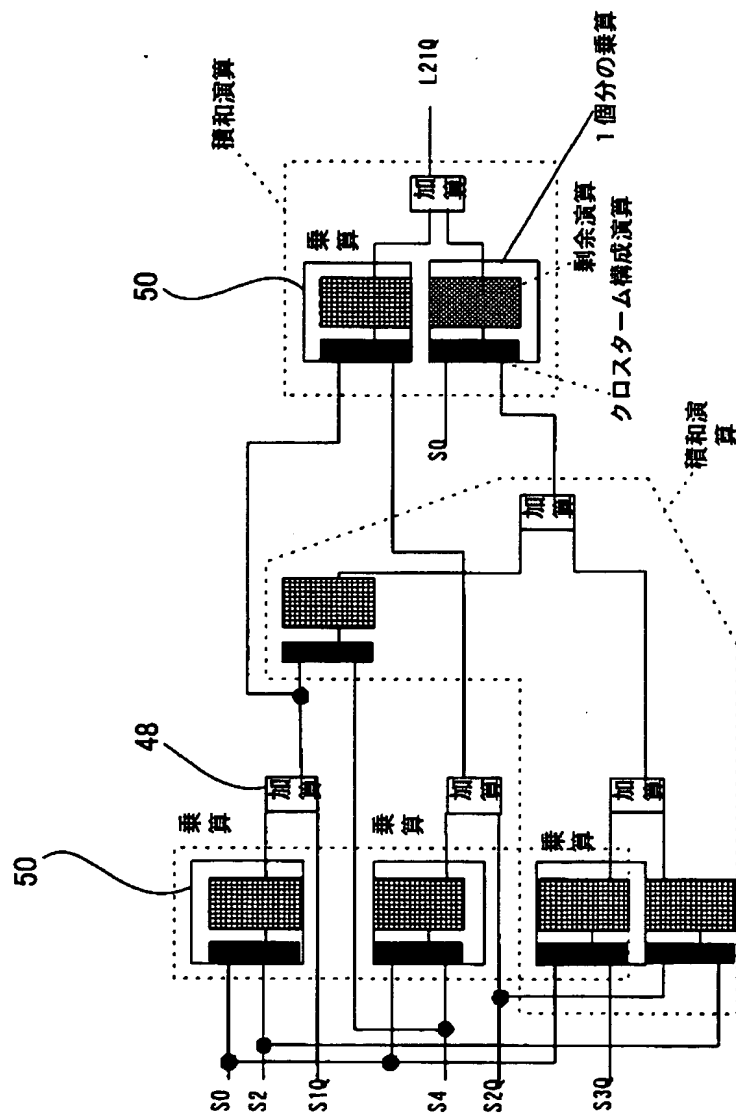
【図 7】



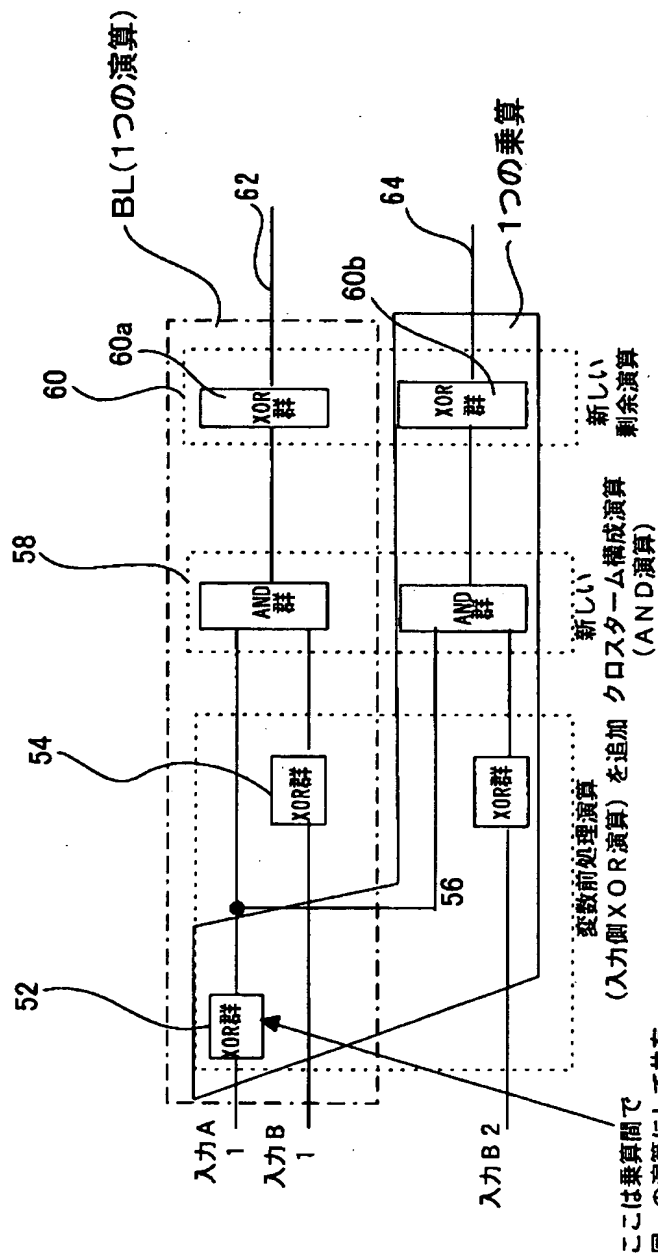
【図 8】



【図9】

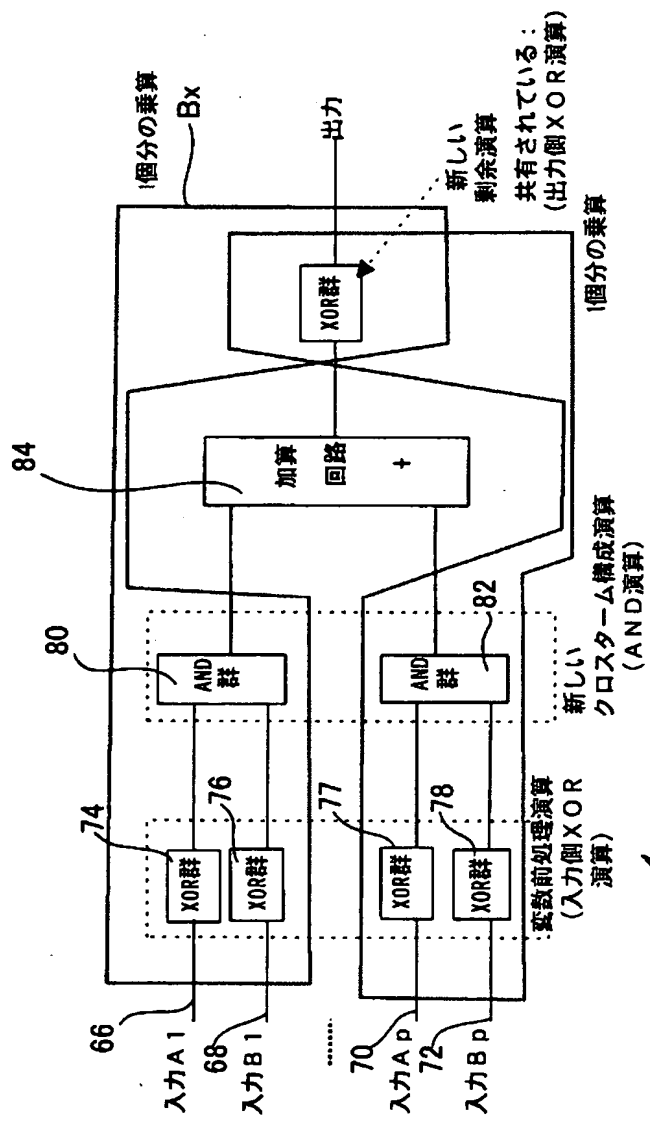


【図10】



(他の共有しないXOR群は乗算ごと異なっても良い)

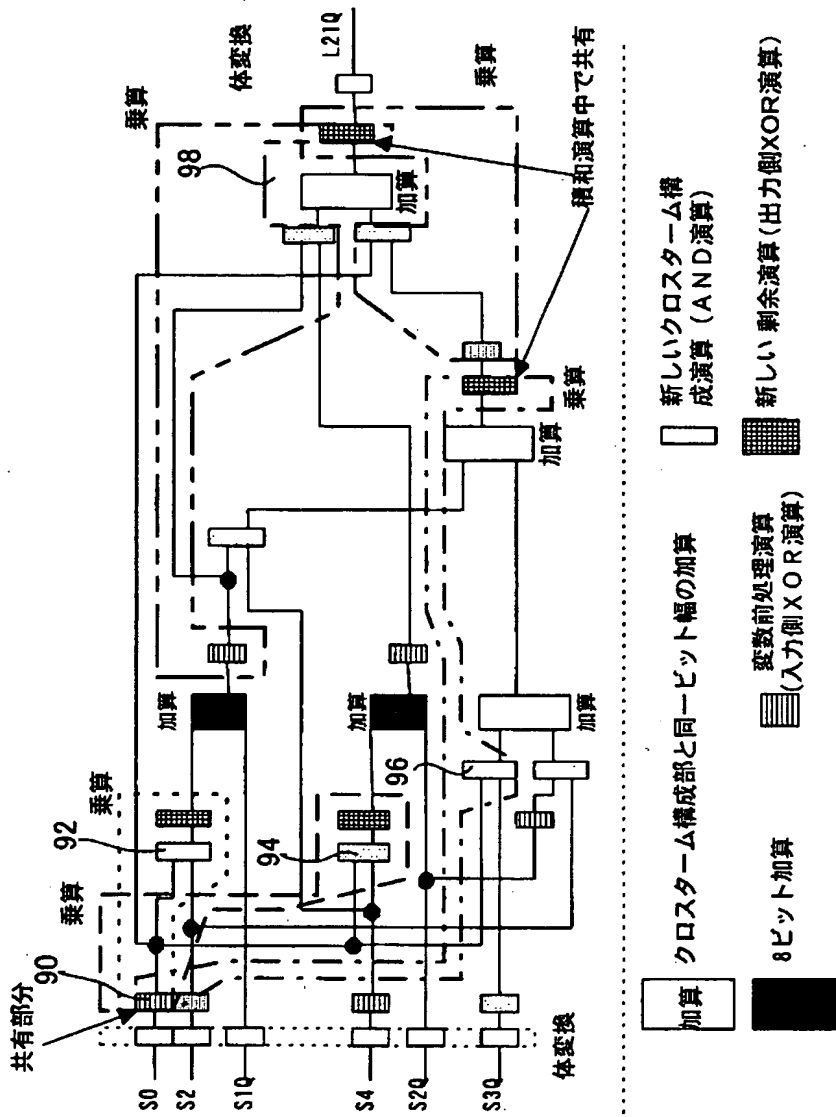
【図 11】



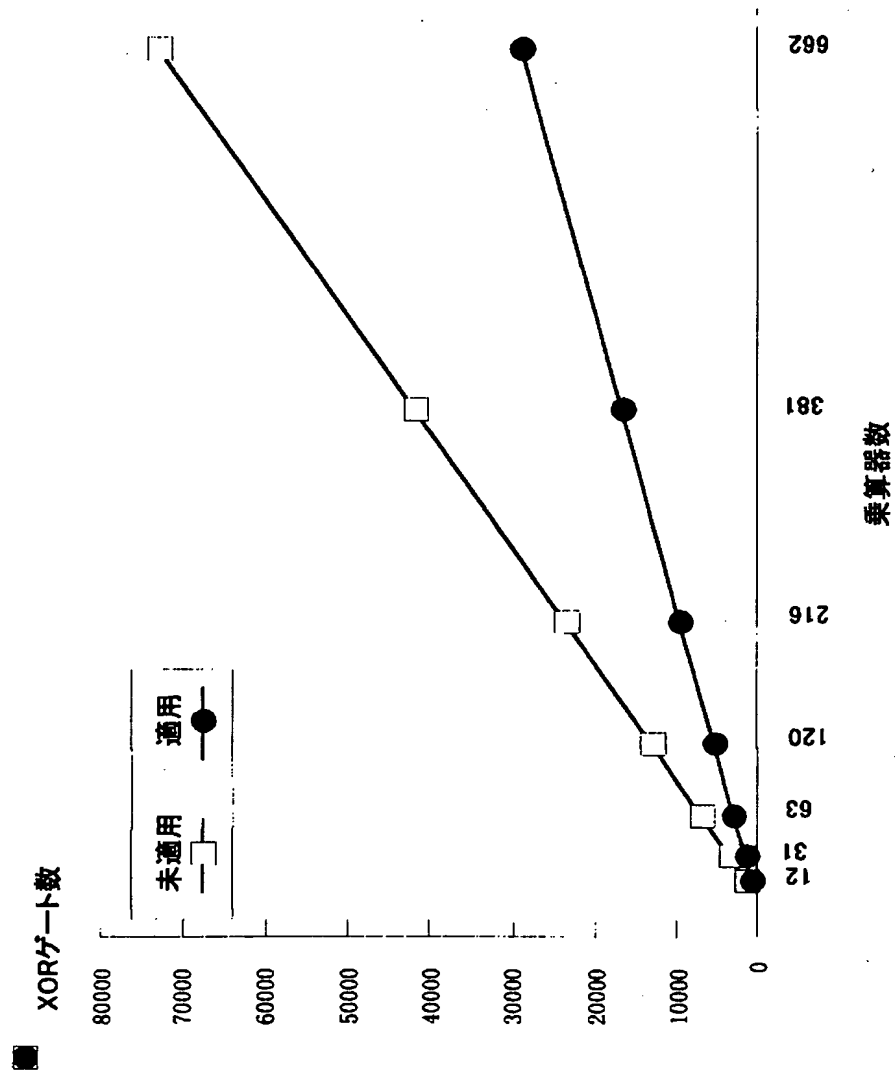
仮にここが共有されなくても、XOR-AND-XOR になる場合がある



【図 12】



【図 1 3】



【図 1 4】

$$\tilde{\Lambda}_1^{(1)} = S_1$$

$$\tilde{\Lambda}_0^{(1)} = S_0$$

$l=2$ の場合

$$\tilde{\Lambda}_2^{(2)} = \tilde{\Lambda}_1^{(1)} S_3 + S_2^2$$

$$\tilde{\Lambda}_1^{(2)} = \tilde{\Lambda}_0^{(1)} S_3 + \tilde{\Lambda}_1^{(1)} S_2$$

$$\tilde{\Lambda}_0^{(2)} = \tilde{\Lambda}_0^{(1)} S_2 + \tilde{\Lambda}_1^{(1)} S_1$$

$l=3$ の場合

$$\tilde{\Lambda}_3^{(3)} = \tilde{\Lambda}_2^{(2)} S_5 + S_1 S_4^2 + S_3^3$$

$$\tilde{\Lambda}_2^{(3)} = \tilde{\Lambda}_1^{(2)} S_5 + \tilde{\Lambda}_2^{(2)} S_4 + S_0 S_4^2 + S_2 S_3^2$$

$$\tilde{\Lambda}_1^{(3)} = \tilde{\Lambda}_0^{(2)} S_5 + \tilde{\Lambda}_1^{(2)} S_4 + \tilde{\Lambda}_2^{(2)} S_3$$

$$\tilde{\Lambda}_0^{(3)} = \tilde{\Lambda}_0^{(2)} S_4 + \tilde{\Lambda}_1^{(2)} S_3 + \tilde{\Lambda}_2^{(2)} S_2$$

$l=4$ の場合

$$\tilde{\Lambda}_4^{(4)} = \tilde{\Lambda}_3^{(3)} S_7 + \tilde{\Lambda}_2^{(3)} S_6^2 + S_4^4 + S_3 S_4^2 S_5 + S_3^2 S_5^2 + S_1 S_5^3$$

$$\tilde{\Lambda}_3^{(4)} = \tilde{\Lambda}_2^{(3)} S_7 + \tilde{\Lambda}_3^{(3)} S_6 + \tilde{\Lambda}_1^{(3)} S_6^2 + S_3 S_4^3 + S_2 S_4^2 S_5 + S_1 S_4 S_5^2 + S_0 S_5^3$$

$$\tilde{\Lambda}_2^{(4)} = \tilde{\Lambda}_1^{(3)} S_7 + \tilde{\Lambda}_2^{(3)} S_6 + \tilde{\Lambda}_3^{(3)} S_5 + \tilde{\Lambda}_0^{(3)} S_6^2 + S_3^2 S_4^2 + S_2 S_4^3 + S_2^2 S_5^2 + S_0 S_4 S_5^2$$

$$\tilde{\Lambda}_1^{(4)} = \tilde{\Lambda}_0^{(3)} S_7 + \tilde{\Lambda}_1^{(3)} S_6 + \tilde{\Lambda}_2^{(3)} S_5 + \tilde{\Lambda}_3^{(3)} S_4$$

$$\tilde{\Lambda}_0^{(4)} = \tilde{\Lambda}_0^{(3)} S_6 + \tilde{\Lambda}_1^{(3)} S_5 + \tilde{\Lambda}_2^{(3)} S_4 + \tilde{\Lambda}_3^{(3)} S_3.$$

【図 1 5】

$$\Lambda_i^{(l)} = \frac{\tilde{\Lambda}_i^{(l)}}{\tilde{\Lambda}_0^{(l)}}, \quad i = 1, \dots, l$$

$$\tilde{\Lambda}_0^{(l)} = \begin{vmatrix} S_0 & \cdots & S_{l-1} \\ \vdots & \ddots & \vdots \\ S_{l-1} & \cdots & S_{2l-2} \end{vmatrix},$$

$$\tilde{\Lambda}_i^{(l)} = \begin{vmatrix} S_0 & \cdots & S_{l-1} \\ \vdots & \ddots & \vdots \\ S_{l-i-1} & \cdots & S_{2l-i-2} \\ S_{l-i+1} & \cdots & S_{2l-i} \\ \vdots & \ddots & \vdots \\ S_l & \cdots & S_{2l-1} \end{vmatrix}, \quad i = 1, \dots, l-1$$

$$\tilde{\Lambda}_l^{(l)} = \begin{vmatrix} S_1 & \cdots & S_l \\ \vdots & \ddots & \vdots \\ S_l & \cdots & S_{2l-1} \end{vmatrix}.$$

【図 1 6】

$$\Gamma_0^{(l+1)} = \begin{vmatrix} s_0 & s_1 & \cdots & s_{l-1} \\ s_1 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ s_{l-1} & \cdots & \cdots & s_{2l-2} \end{vmatrix},$$

$$\Gamma_i^{(l+1)} = \begin{vmatrix} s_0 & \cdots & s_{l-1-i} & s_{l+1-i} & \cdots & s_l \\ \vdots & & \vdots & \vdots & & \vdots \\ s_{l-1-i} & \cdots & s_{2l-1-i} & s_{2l-i} & \cdots & s_{2l-1-i} \\ s_{l+1-i} & \cdots & s_{2l-i} & s_{2l+1-i} & \cdots & s_{2l+1-i} \\ \vdots & & \vdots & \vdots & & \vdots \\ s_l & \cdots & s_{2l-1-i} & s_{2l+1-i} & \cdots & s_{2l} \end{vmatrix} \quad i = 1, \cdots, l-1$$

$$\Gamma_l^{(l+1)} = \begin{vmatrix} s_2 & \cdots & \cdots & s_{l+1} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ s_{l+1} & \cdots & \cdots & s_{2l} \end{vmatrix}.$$

【図 1 7】

$$\Gamma_0^{(1)} = 1$$

$$\Gamma_0^{(2)} = S_0$$

$$\Gamma_1^{(2)} = S_2$$

$$\Gamma_0^{(3)} = S_0 S_2 + S_1^2$$

$$\Gamma_1^{(3)} = S_0 S_4 + S_2^2$$

$$\Gamma_2^{(3)} = S_2 S_4 + S_3^2$$

$$\Gamma_0^{(4)} = \Gamma_0^{(3)} S_4 + S_0 S_3^2 + S_2^3$$

$$\Gamma_1^{(4)} = \Gamma_0^{(3)} S_6 + S_0 S_4^2 + S_2 S_3^2$$

$$\Gamma_2^{(4)} = \Gamma_1^{(3)} S_6 + S_0 S_3^2 + S_4 S_3^2$$

$$\Gamma_3^{(4)} = \Gamma_2^{(3)} S_6 + S_2 S_3^2 + S_4 S_4^2$$

$$\Gamma_0^{(5)} = \Gamma_0^{(4)} S_6 + \Gamma_0^{(3)} S_3^2 + \Gamma_1^{(3)} S_4^2 + \Gamma_2^{(3)} S_3^2$$

$$\Gamma_1^{(5)} = \Gamma_0^{(4)} S_8 + \Gamma_0^{(3)} S_6^2 + \Gamma_1^{(3)} S_3^2 + \Gamma_2^{(3)} S_4^2$$

$$\det 03 = S_0 S_6 + S_3^2$$

$$\det 24 = S_2 S_6 + S_4^2$$

$$\Gamma_2^{(5)} = \Gamma_1^{(4)} S_8 + \Gamma_0^{(3)} S_7^2 + \det 03 \cdot S_3^2 + \det 24 \cdot S_4^2$$

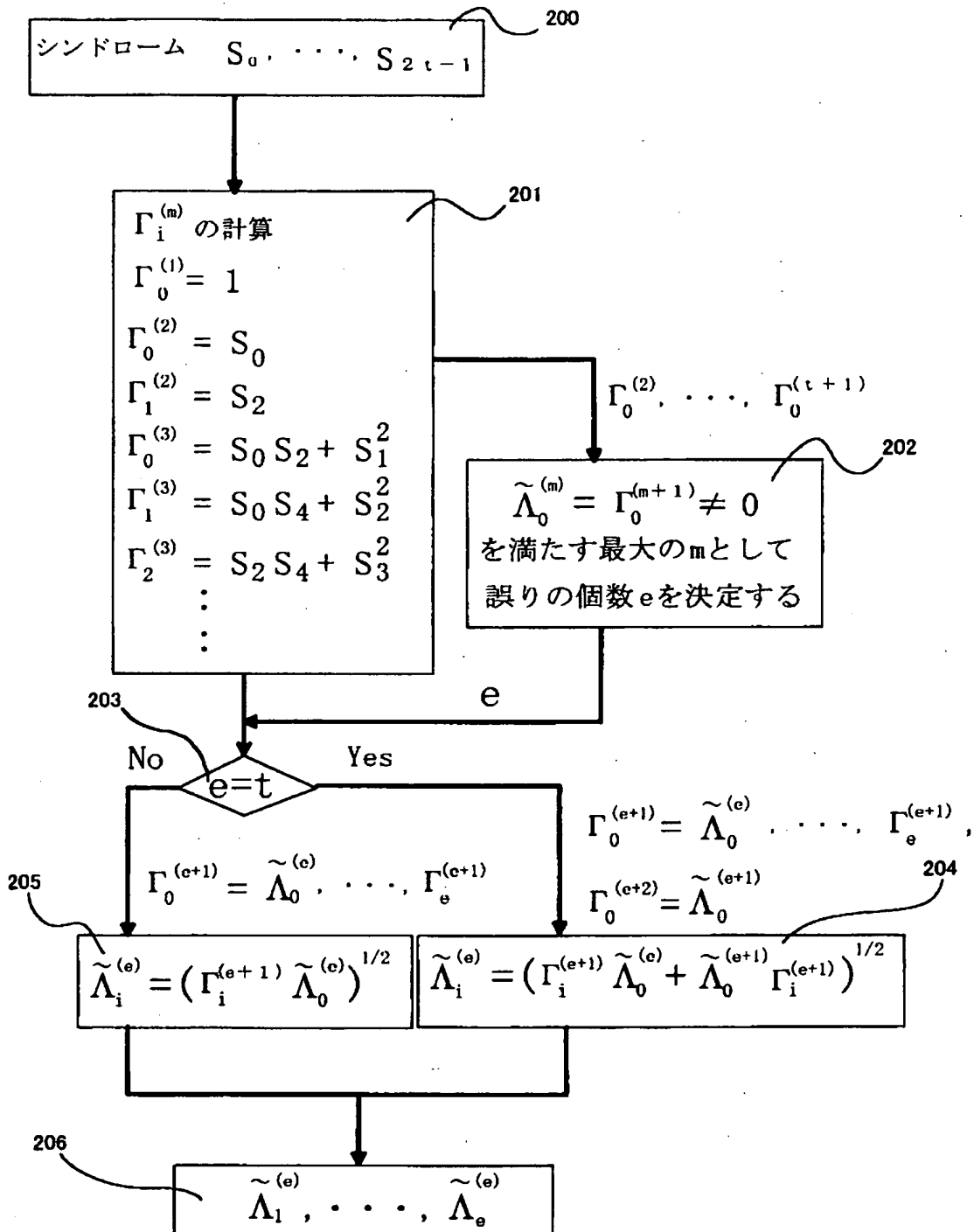
$$\det 45 = S_4 S_6 + S_5^2$$

$$\Gamma_3^{(5)} = \Gamma_2^{(4)} S_8 + \Gamma_1^{(3)} S_7^2 + \det 03 \cdot S_6^2 + \det 45 \cdot S_4^2$$

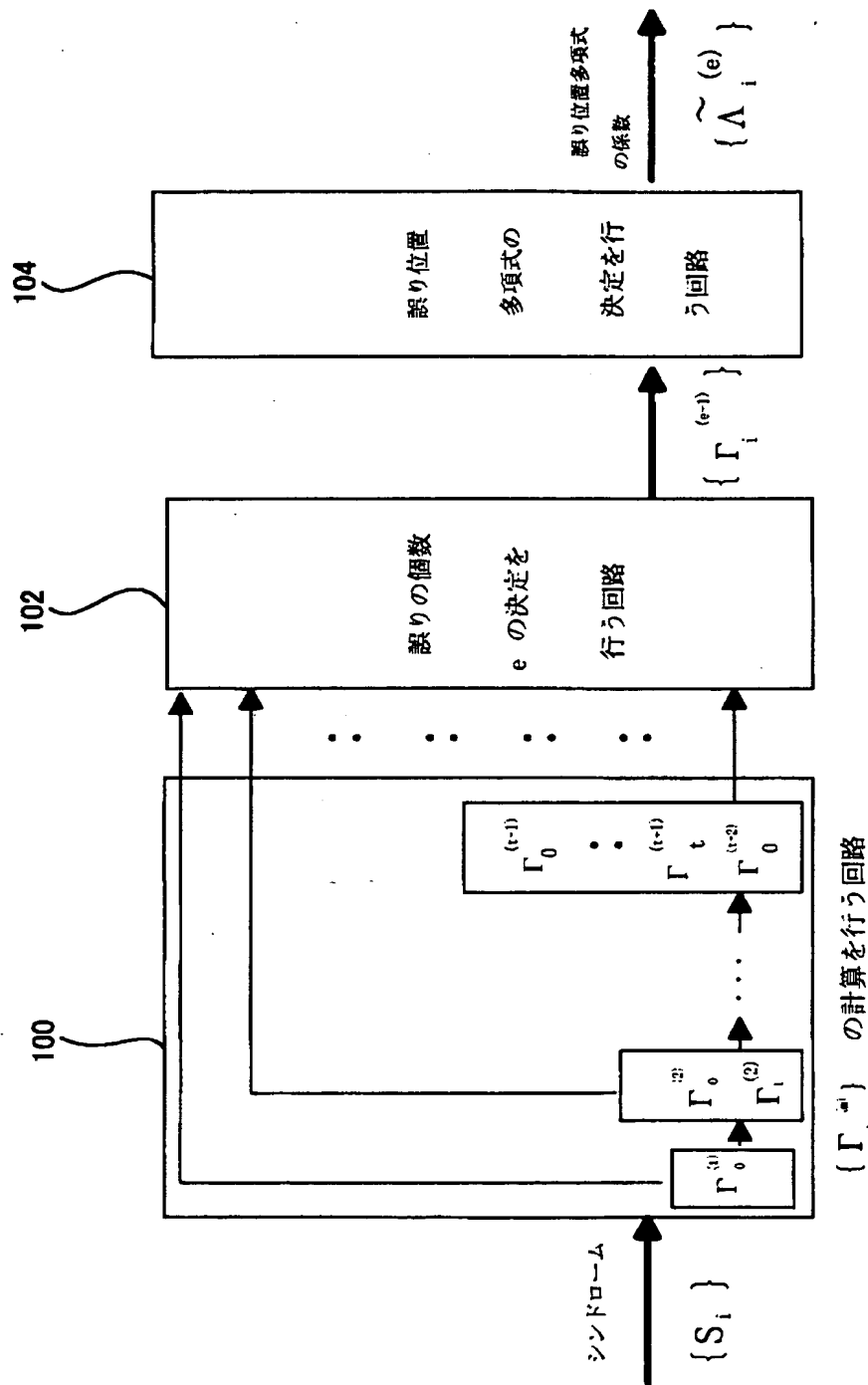
$$\Gamma_4^{(5)} = \Gamma_3^{(4)} S_8 + \Gamma_2^{(3)} S_7^2 + \det 24 \cdot S_6^2 + \det 45 \cdot S_3^2$$

$$\Gamma_0^{(6)} = \Gamma_0^{(5)} S_8 + \Gamma_0^{(4)} S_7^2 + \Gamma_1^{(4)} S_6^2 + \Gamma_2^{(4)} S_3^2 + \Gamma_3^{(4)} S_4^2$$

【図 18】

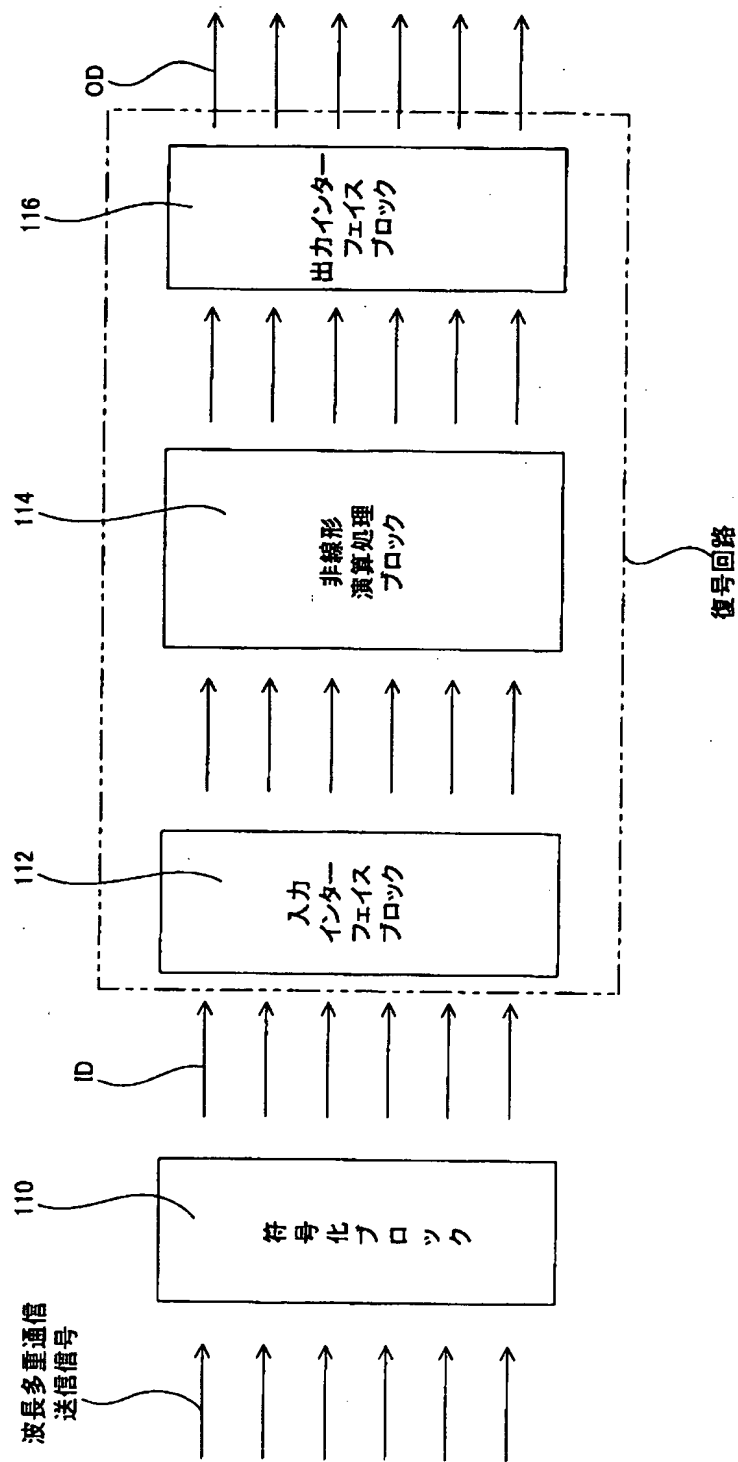


【図 19】





【図 20】



【書類名】 要約書

【要約】

【課題】 復号回路、復号装置、復号方法、および半導体デバイスを提供する。

【解決手段】 デジタル信号の誤りを訂正するために使用される復号回路であって、該復号回路は、符号化された入力デジタル信号 I D が符号のインターリーブに対応して並列して入力される入力部 1 0 と、入力部 1 0 からのインターリーブされた符号語間でシリアル化されたデータ出力をそれぞれ処理する誤り位置多項式係数算出部 1 8 と、誤り値多項式係数算出部 2 0 とを含む処理部 1 2 と、処理部 1 2 からの出力と入力デジタル信号 I D とが入力され、処理部 1 2 からの出力と、入力デジタル信号とから、ガロア体における線形演算で誤りが訂正された出力デジタル信号 O D を符号のインターリーブに対応して並列に出力する出力部 1 4 とを備える復号回路を含む。

【選択図】 図 6

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 2000年 5月16日  
[変更理由] 名称変更  
住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)  
氏 名 インターナショナル・ビジネス・マシーンズ・コーポレーション